Online Ethiopic Handwriting Character Recognition System (OEHCRS): a Hybrid Approach using Discrete HMM over Structural Primitives

Yenealem Nibret
HiLCoE, Computer Science Programme, Ethiopia
yenealemn@gmail.com

Wondossen Mulugeta
HiLCoE, Ethiopia
School of Information Science, Addis Ababa University, Ethiopia
wondisho@yahoo.com

Abstract

The objective of the research is to explore, analyze, and design online handwriting character recognition system for Ethiopic script. To meet the set objective, two fundamental pattern recognition tasks: pattern description and pattern classification are proposed, designed and incorporated in the system. The pattern description task contains preprocessing and feature extraction activities in order to describe the handwriting with a uniform format of description; and the pattern classification task is done by using HMM (Hidden Markov Model) classifier. A corpus of UNIPEN has been used for the Model training. Using the file, HMMs for characters of the Alphabet have been built using Baum-Welch training on 70% of the corpus while the remaining is kept for testing. To search a matching model for a handwriting easily, the HMMs are organized on a three level hierarchy, and Viterbi computations are conducted to reach a most likelihood Model. According to the test to evaluate the models, promising results have been achieved (average of 99.12%, 97.75 %, 93.99% for top level, middle level, and bottom level HMMs, respectively).

Keywords: Online Character Recognition System; Ethiopic; HMM; Pattern Description; Pattern Classification

1. Introduction

Ethiopic script was developed as a writing system of a Semitic language called Geez. Historical Geez inscriptions are dated back to 5 B.C. Nowadays, the Geez language is used only for liturgical purposes in Ethiopian and Eritrean Orthodox churches. Though Geez has ceased in vernacular speech, the script is used for writing systems of different languages in Ethiopia and Eritrea. Amharic, Agew, Argoba, Bilen, Gurage (Cheha), Tigre, and Tigrigna are some of the languages which use Ethiopic script for writing. The Alphabet and character of Ethiopic script are referred to as 'fidelgebeta' /ፊደልገበታ/ and 'fidel' /ፊደል/ respectively. Characters in the Alphabet have been extending in number for many years. In 2009, the Alphabet had been standardized to have 435 characters [2].

Ethiopic Alphabet is organized in three main classes. The first class contains 238 different characters grouped into 34 sets. Each set consists of 7 ordered characters and has a representative base sound. The base sounds of the set being combined with the vowels of the Alphabet ('ä', 'u', 'i', 'a', 'e', 'ə' and 'o') form the seven characters (syllographs).

The second class contains labialized characters of the first class's base sounds. However, some characters in the first class do not have correspondent labialized character in the second class. This is due to absence of words with these sounds in the languages.

The third class holds a number of derived characters from the first class.

Figure 1 shows characters of first and second classes of the Alphabet. This research mainly targets characters of the first and second classes which are 265 in total. However, due to structural similarity of the characters, methods used for recognition of the 265 characters can also be used for the remaining ones.

This paper explores, analyzes, and designs online handwriting character recognition system for Ethiopic script.

Base sounds	Vowels ä	u	i	A	e	æ	O
h	ሀ/hä/	ሁ/hu/	ሂ/hi/	ሃ/ha/	ሄ/he/	ህ/hæ/	ሆ/ho/
l	ለ/lä/	ሉ/lu/	ሊ/li/	ላ/la/	ሌ/le/	ል/læ/	ሎ/lo/
h	ሐ/hä/	ሑ/hu/	ሒ/hi/	ሓ/ha/	ሔ/he/	ሕ/hæ/	ሖ/ho/
m	መ/mä/	ሙ/mu/	ሚ/mi/	ማ/ma/	ሜ/me/	ም/mæ/	ሞ/mo/
s	ሠ/sä/	ሡ/su/	ሢ/si/	ሣ/sa/	ሤ/se/	ሥ/sæ/	ሦ/so/
r	ረ/rä/	ሩ/ru/	ሪ/ri/	ራ/ra/	ሬ/re/	ር/ræ/	ሮ/ro/
s	ሰ/sä/	ሱ/su/	ሲ/si/	ሳ/sa/	ሴ/se/	ስ/sæ/	ሶ/so/
Š	ሸ/Šä/	ሹ/Šu/	ሺ/Ši/	ሻ/Ša/	ሼ/Še/	ሽ/Šæ/	ሾ/Šo/
k	ቀ/kä/	ቁ/ku/	ቂ/ki/	ቃ/ka/	ቄ/ke/	ቅ/kæ/	ቆ/ko/
b	በ/bä/	ቡ/bu/	ቢ/bi/	ባ/ba/	ቤ/be/	ብ/bæ/	ቦ/bo/
t	ተ/tä/	ቱ/tu/	ቲ/ti/	ታ/ta/	ቴ/te/	ት/tæ/	ቶ/to/
ç	ቸ/çä/	ቹ/çu/	ቺ/çi/	ቻ/ça/	ቼ/çe/	ች/çæ/	ቾ/ço/
h	ኀ/hä/	ኁ/hu/	ኂ/hi/	ኃ/ha/	ኄ/he/	ኅ/hæ/	ኆ/ho/
n	ነ/nä/	ኑ/nu/	ኒ/ni/	ና/na/	ኔ/ne/	ን/næ/	ኖ/no/
ň	ኘ/ňä/	ኙ/ňu/	ኚ/ňi/	ኛ/ňa/	ኜ/ňe/	ኝ/ňæ/	ኞ/ňo/
a	አ/aä/	ኡ/au/	ኢ/ai/	ኣ/aa/	ኤ/ae/	እ/aæ/	ኦ/ao/
k	ከ/kä/	ኩ/ku/	ኪ/ki/	ካ/ka/	ኬ/ke/	ክ/kæ/	ኮ/ko/
ǩ	ኸ/ǩä/	ኹ/ǩu/	ኺ/ǩi/	ኻ/ǩa/	ኼ/ǩe/	ኽ/ǩæ/	ኾ/ǩo/
w	ወ/wä/	ዉ/wu/	ዊ/wi/	ዋ/wa/	ዌ/we/	ው/wæ/	ዎ/wo/
a	ዐ/aä/	ዑ/au/	ዒ/ai/	ዓ/aa/	ዔ/ae/	ዕ/aæ/	ዖ/ao/
z	ዘ/zä/	ዙ/zu/	ዚ/zi/	ዛ/za/	ዜ/ze/	ዝ/zæ/	ዞ/zo/
ž	ዠ/žä/	ዡ/žu/	ዢ/ži/	ዣ/ža/	ዤ/že/	ዥ/žæ/	ዦ/žo/
y	የ/yä/	ዩ/yu/	ዪ/yi/	ያ/ya/	ዬ/ye/	ይ/yæ/	ዮ/yo/
d	ደ/dä/	ዱ/du/	ዲ/di/	ዳ/da/	ዴ/de/	ድ/dæ/	ዶ/do/
j	ጀ/jä/	ጁ/ju/	ጂ/ji/	ጃ/ja/	ጄ/je/	ጅ/jæ/	ጆ/jo/
g	ገ/gä/	ጉ/gu/	ጊ/gi/	ጋ/ga/	ጌ/ge/	ግ/gæ/	ጎ/go/
ţ	ጠ/ţä/	ጡ/ţu/	ጢ/ţi/	ጣ/ţa/	ጤ/ţe/	ጥ/ţæ/	ጦ/ţo/
ć	ጨ/ćä/	ጩ/ću/	ጪ/ći/	ጫ/ća/	ጬ/će/	ጭ/ćæ/	ጮ/ćo/
ṕ	ጰ/ṕä/	ጱ/ṕu/	ጲ/ṕi/	ጳ/ṕa/	ጴ/ṕe/	ጵ/ṕæ/	ጶ/ṕo/
ş	ጸ/şä/	ጹ/şu/	ጺ/şi/	ጻ/şa/	ጼ/şe/	ጽ/şæ/	ጾ/şo/
ş	ፀ/şä/	ፁ/şu/	ፂ/şi/	ፃ/şa/	ፄ/şe/	ፅ/şæ/	ፆ/şo/
f	ፈ/fä/	ፉ/fu/	ፊ/fi/	ፋ/fa/	ፌ/fe/	ፍ/fæ/	ፎ/fo/
p	ፐ/pä/	ፑ/pu/	ፒ/pi/	ፓ/pa/	ፔ/pe/	ፕ/pæ/	ፖ/po/
v	ቨ/vä/	ቩ/vu/	ቪ/vi/	ቫ/va/	ቬ/ve/	ቭ/væ/	ቮ/vo/

(a)

ሏ/lwa/	ሟ/mwa/	ሯ/rwa/	ሷ/swa/	ሿ/Šwa/	ቋ/kwa/	ቈ/kwa/	ቧ/bwa/	ቷ/twa/
ቿ/çwa/	ኋ/hwa/	ኗ/nwa/	ኟ/ňwa/	ኳ/kwa/	ዟ/zwa/	ጧ/žwa/	ጓ/gwa/	ጇ/dwa/
ጇ/jwa/	ጧ/ţwa/	ጯ/ćwa/	ፏ/śwa/	ፏ/fwa/	ዎ/ǩwo/	ኰ/kwo/	ጐ/gwo/	ኧ/awo/

(b)

Figure 1: (a) First Class and (b) Second Class Characters in Ethiopic Alphabet

2. Literature Review

Since the first ordered characters of Ethiopic Alphabet are bases for shapes of other characters, they are referred to as form-characters [1]. The formation of second and third class characters follows application of modifying effects such as appendage, break (zigzag), loop, and vertical standing on the form characters. However, these modifying effects do not eliminate form-characters' shape completely. This makes derived characters of a form-character contain common structural patterns. Hence, the structural (shape) variations among characters under a form-character are smaller than the variations among different form-characters. In addition, the characters

share some common properties according to the order they have and/or the class they belong to. In the first class, there are similarities among characters on the applied modifying effects at each particular order of the Alphabet despite existence of some exceptions. For instance, second order characters use appendage at their right middle positions.

The following list provides brief descriptions on the similarities of distinct characters according to the applied modifying effects.

- All the 5th ordered characters add loops at their right bottom part of the Alphabet, except the characters 'ረ ' and 'ፈ '.
- The 2nd and 3rd ordered characters do have right appendage at their right middle section and right bottom appendage, respectively. In the order, ('ሩ' and 'ፉ') from the 2nd ordered characters and ('ሪ' and 'ፊ') form the 3rd ordered ones are exceptions.
- Characters in the 4th order have shorten left and center (if any) standing line(s) if their form-characters have two or more standings such as 'በ', 'ሰ', and 'ሸ'. They have fore-slash line at their bottom if their form-characters have one standing such as 'ተ', 'ቀ', and 'ፐ'- except 'ነ' and 'ኘ' and a vertical stand at their bottom right (center) if their form-characters do not have standing. However, 'ና', 'ኟ', and 'ራ' are exceptions.
- The 7th ordered characters are formed with: shortening the right and center (if any) standing line(s) if their form-characters have two or more standings; attaching loop at the top center (right) section of their form-characters if their standing is a single vertical line and adding fore-slash or vertical lines at their bottom left if their form-characters have not standing. However; 'ሆ', 'ሎ', and 'ፖ' are exceptions.
- Left appendages, right appendages, line breaks and top back-slashes are basic features of 6th ordered characters; except 'ል', 'ር', 'ፍ', 'ዕ', and 'ፅ'.
- Characters of the second class are formed with application of horizontal appendage either at the 1st or 4th ordered characters of their order sets.

3. The Proposed Solution

The design of Online Ethiopic Handwriting Character Recognition System (OEHCRS) includes principal components that are believed to address basic pattern recognition functions for the intended assignment. The design of our approach bases on the tasks (pattern description and classification) and their elementary activities inclusively. Under pattern description task, preprocessing and feature extraction activities are incorporated to pave the way for defining pattern representation for raw pen data into vector form while HMMs development and organization rules are parts of the classification task. Figure 2 is block diagram of OEHCRS. Next, the activities of the tasks are presented.

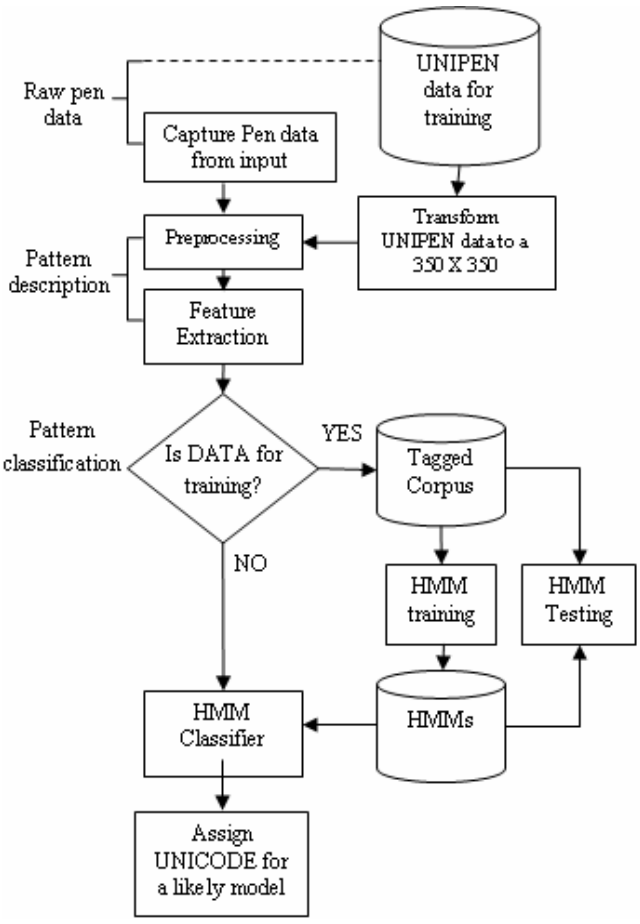

Figure 2: Architectural Design of the OHCRS

3.1 Preprocessing Activities

Preprocessing activities remove noises of handwritings that occur during handwriting that might reduce recognition and make the data ready for feature extraction. In our case, the activities include the following:

- *Removal of redundant coordinates*: since they do not have useful information for recognition during data capturing.
- *Grouping coordinates*: Each captured coordinate, except the first coordinate of a stroke, is assigned orientation value according to its |sin θ| value with respect to its immediate preceding coordinates. The orientations are classified into three as Horizontal, Slash, and Vertical for Sin values [0, 0.5], (0.5, 0.866), and [0.866, 1.0]. The three orientations have equal range of angles as shown in Figure 3.
- *Combining strokes:* Two or more formatted strokes are grouped into one group when they are closer enough. This lets primitive extraction work easier, especially for loop detection and extraction.

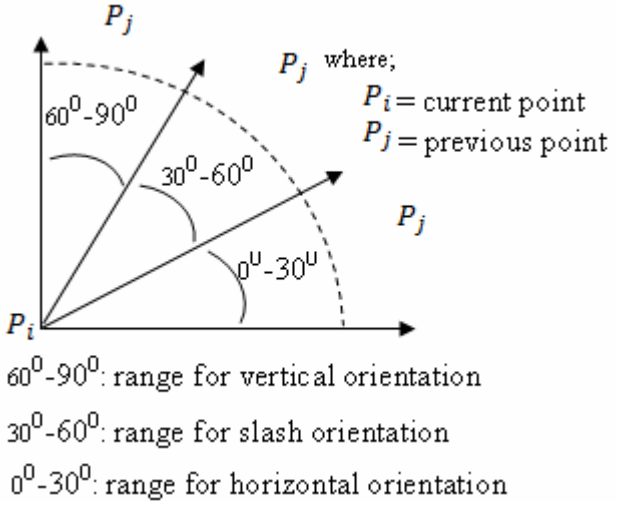

Figure 3: Ranges of Degree of the Orientations

3.2 Feature Extraction

Using preprocessed handwritings, feature extraction activities derive primitives and generates a feature vector for classification purpose. The activities are the following.

- *Loop detection:* detect loops based on intersection of projections at the horizontal and vertical axes of the spatial alignment of strokes. Naturally, a loop can only be made with three or more stroke combinations and the sub-strokes make two different intersection types (horizontal and vertical intersections) at their axes (x-axis and y-axis) projections geometrically. Figure 4 illustrates the detection approach using the Ethiopic character '*ውመ*' /mä/.
- *Defining stroke set of ligated loops*: Identify sharable sub-strokes between ligated loops and prepare separate sub-stroke sets for two loops.
- *Defining boundaries of loops*: Boundary of each loop is defined with minimum and maximum values of coordinates of the identified set of sub-strokes.
- *Removing noise strokes and sub-strokes*: After boundaries of each loop are identified and their spatial coverage is set, the constituent strokes and sub-strokes are removed in order to avoid duplication of information. In addition, cuts of sub-strokes and very short stroke parts to represent a primitive are removed from the set. So, loops and other lines (Horizontal, Slash, and Vertical) are set as primitives of the handwriting.
- *Zoning primitives*: Identified primitives are allocated in 3X3 spatial zones which are delimited according to the structure of the character by which top and bottom zones are assigned 20% of the vertical dimension and the horizontal dimension is equally divided into 3 zones.
- *Coding primitives*: Each primitive is assigned code according to its primitive type and spatial zone.

As a result of the above activities, at most 9 sets of primitives are generated with their zone labels.

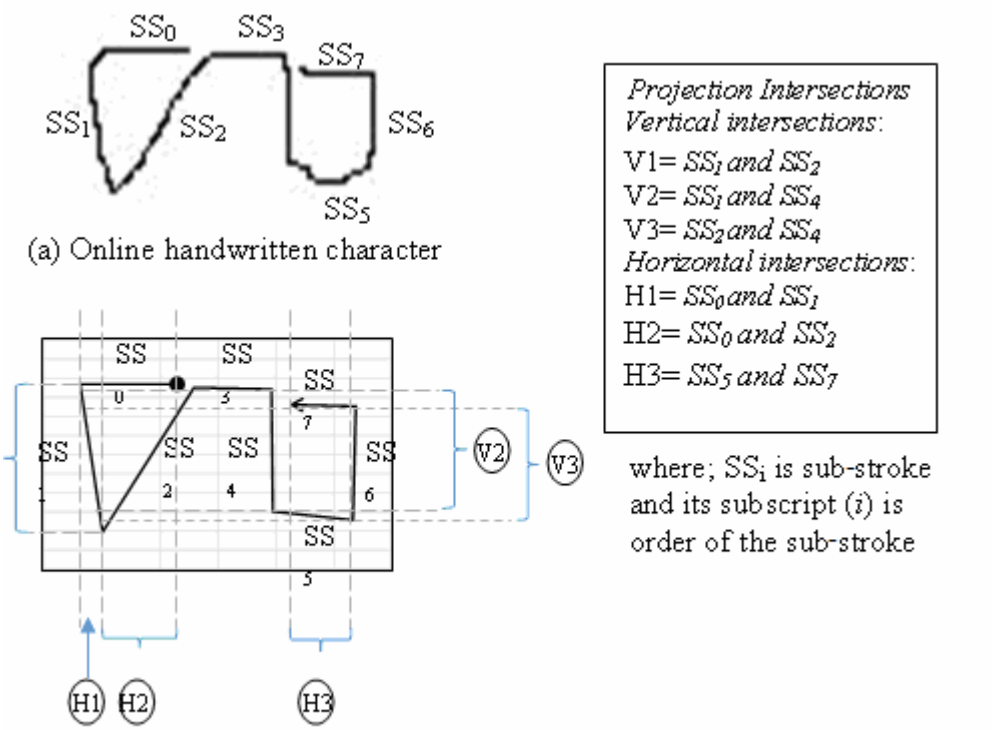

Figure 4: Projection Intersections in a Handwritten Looped Ethiopic Character 'ሙ'/ mä/.

3.3 Classification Activities

During classification of the handwriting, two principal tasks are done: HMM development and hierarchical organization of HMMs.

- *HMM development*: For each character/model holder of the system, using Baum-welch training on 70% of the corpus, HMMs are developed. In the definition of each HMM, primitives are treated as observations (O) while their spatial zones are states (S). Primitives: Vertical line, Slash line, Horizontal line, and Loop are represented with codes: '1', '2', '3', and '4', respectively. The 9 zones are represented with labels from left to right and top to bottom of the divisions as 'FT', 'SD', 'TD', FR', 'FV', SX', 'SN', 'EG', and 'NI'.
- *Hierarchical organization of HMMs*: HMMs are also developed for clusters of handwritings according to structural similarity between the characters. The clusters' HMMs and characters' HMMs are arranged in three level hierarchically as Top, Middle, and Bottom levels. The Top, Middle, and Bottom levels contain 8, 34, and 265 HMMs, respectively. The top 8 HMMs are formed using rules set (see Table 1) according to the structural similarities among the form-characters and the 34 HMMs are created based on the fact that structural similarities among the 34 form-characters and their derivatives and the 265 HMMs represent individual characters of the script.
- *Recognition*: This is done by searching the most likelihood of a pattern description from the tree of HMMs and picking the equivalent Unicode of the found HMM. The searching process inputs a newly recognizable handwriting with appropriate observations and states sequence, and finds ($Max\left[P_i(O,S)\middle|\lambda_i\right]_j$) which is the most joint likelihood of the observations (O) and states (S) from HMMs ($\lambda_i; i = 0,1,2,3 \ldots n$) on a search line of the tree at each level (j) of the hierarchy. Computation of maximum joint probability of observations-states sequence against HMMs is done with Viterbi algorithm [3].

In order to return character symbols for the handwriting, the 265 HMMs with their respective Unicode values are listed in a dictionary so that the selected character's HMM at third level of the hierarchy is mapped to a corresponding Unicode using the dictionary. Algorithm 1 shows searching and finding a maximum likely HMM model from the hierarchy of HMMs for the handwriting.

```
Function SearchFindHMM({O,S}, model)
{
   h ← 0
   while (h < 3)
   {
      i ← o
      log↓Probabilities←[]
   }
   while (i < length of model)
   {
      log↓Probabilities[i]← [logP]
         ↓i(O,S|λ↓i)
      i ← i+1
      MaxLikModelIndex ←
         Index(Max(log↓Probabilities))
   }
   model ← model[MaxLikModelIndex]
  h ← h + 1
   return model
}
```

Algorithm 1: Searching Maximum Joint Log Probability of (O, S) from HMMs' Tree

Table 1: Clustering Rules and Governed Form-Characters

Rule	*Form-characters*
Left vertical line and bottom horizontal line	ሀ, ሠ, ረ, ፈ
Two or more vertical lines standing with a middle horizontal line	ለ, ሐ, ዘ, አ
Bottom loops	ሟ, ወ, ዐ, ፏ, ፀ
Two or more vertical lines standings with top horizontal line	ሰ, ቨ, ከ, ጠ
Two or more vertical lines standings with top appendage	ሸ, ኸ, ዠ, ቨ
horizontal line	
A central vertical line standing with a middle or top horizontal line	ቀ, ተ, ቸ, ፐ
A right vertical line with a middle or top horizontal line	ነ, ኘ, ጎ, ገ
Top loops	የ, ደ, ጀ, ጸ, ጸ

4. Experimentation

Experiments have been conducted to measure performance of the loop detection technique and accuracies of the discrete HMMs which use primitives' sequences fundamentally. For experimentation, UNIPEN data that was collected from 65 writers in [2] has been used. For all the programming works, Python 2.7.3 is used, Python modules: Tkinter 8.5 and Numpy 1.8 are applied for graphical user interface (GUI) construction and some statistical works respectively, and Natural Language Tool Kit (NLTK 2.0) is also incorporated for HMM training and model development.

Evidently, the pen data (UNIPEN) has up to 12 points difference (gap) between consecutive coordinates. The gap causes to generate misleading primitives and affects the pre-processing tasks. In order to minimize the gap and reduce the number of misleading primitives, the gap should be reduced in size or interpolation should be done. In our case, each pen data has been transformed into a new set of coordinates by reducing its original size (590 X 590) in half so that without additional interpolation task, losing original functions, and distorting shapes of handwriting, coordinates have compacted and become suitable for pre-processing and feature extraction works and be retraceable on 350X 350 writing area of our prototype.

At description task, identification capability of loop extractor is tested. In the process, gap filling between combinable strokes or sub strokes is seen as a considerable factor and thresholds have been set and results are measured since the number of loops at characters is known. It is also seen that a threshold of 10 points gives a better result of precision (59.32%),

recall (49.46%), and F-measure (53.94%) values. Table 2 shows the result with threshold of 10 points using 17031 handwritings according to class and order of the characters.

Table 2: Accuracy of Loop Detector with Threshold of 10 Points

Result	*1st Class*								
	1st order	*2nd order*	*3rd order*	*4th order*	*5th order*	*6th order*	*7th order*	*2nd class*	*Total*
True Positive	321	315	437	336	1217	379	483	496	3984
False Positive	314	399	297	461	0	480	286	495	2732
False Negative	501	449	326	429	967	448	736	215	4071
True Negative	1043	1022	1125	961	0	875	679	539	6244
Total number of handwriting files used 17031									
Precision: 59.32%, Recall: 49.46% , F-measure: 53.94%									

During observation coding, confusions are seen in HMMs due to prevalence of identical primitives on different zones. To avoid this, observations at the 9 zones are labelled with prefixes 'Pa', 'Pb', 'Pc', 'Pd', 'Pe', 'Pf', 'Pg', 'Ph', and 'Pi' orderly and in effect, confusions are highly minimized.

Lastly, the models' accuracies are measured using 30% of the corpus and average accuracies of 99.17%, 97.75%, and 94% are achieved for Top, Middle, and Bottom level HMMs, respectively.

5. Conclusion and Future Work

The performance of loop detector has been tested with the prevalence of loops applying the loop detector algorithm in the feature description task and has resulted in an average of 0.5 for precision, recall, and F-measures at the best case threshold value (10 points). The result, though it seems low in accuracy, is a good achievement because accuracy results at primitive level of a pattern recognizer are always poor as compared to the shell recognition system [4, 5]. However, improving the detector's performance provides better feature extraction ability and increases recognition accuracy. The other critical task of the recognizer, classification, contains 307 HMMs which are arranged hierarchically in three levels. Average sizes of the training corpus per model at Top, Middle, and Bottom are 1489, 350, and 65, respectively. Similarly, the accuracies have dropped down from 99.16% to 93.99%. Hence, differences in accuracy of the models across the three hierarchical levels is directly associated with size of the training corpus. Hence, the model development needs additional handwriting data to train and equip HMMs with various observation-to-state combinations from the targeted class of character(s).

References

[1] A. Shimeles, "Online Handwriting Recognition for Ethiopic Characters for Ethiopic Characters", Addis Ababa University, 2005.

[2] Yaregal Assabie and J. Bigun, "Online Handwriting Recognition of Ethiopic Script," Halmstad University, 2009, pp. 153–158.

[3] R. Sramek, "The on-line Viterbi Algorithm," Comenius University, Bratislava, 2007.

[4] A. Graves, "Supervised Sequence Labelling with Recurrent Neural Networks", Springer: Studies in Computational Intelligence, 2012.

[5] Sunil Kumar Kopparapu and L. Vl, "Online Handwritten Devanagari Stroke Recognition Using Extended Directional Features," in 8th International conference on Signal Processing and Communication Systems, Gold Cost, 2015.