"Data is like a fingerprint, unique and valuable, requiring careful handling and pre-processing"

- Bernard Marr

Chapter 3

DATA COLLECTION, DIGITIZATION AND PRE-PROCESSING

The recognition of offline Devanagari handwritten words involves three main preliminary steps namely data collection, digitization and pre-processing. This chapter provides an insight of the tasks involved in collecting data for offline handwritten Devanagari words, the digitization process and the pre-processing operations. Section 3.1 presents the details of the data collection process, Section 3.2 outlines the digitization phase and Section 3.3 focuses on the pre-processing phase of the offline HWR system. In Section 3.4, performance analysis of various thinning algorithms for offline handwritten words have been presented. Finally, Section 3.5 offers a summary of the entire chapter.

3.1 DATA COLLECTION

For performing experiments and facilitating comparisons among various methodologies, a benchmark database is essential. However, there was no publicly accessible benchmark database/corpus for the Devanagari script. Therefore, in order to recognize offline handwritten words in Devanagari script, a database/corpus was developed. For this work, a database/corpus of 48,000 handwritten Devanagari words (120 town/city names written by 400 writers each = 48,000 words) have collected from writers with variation in their age, qualification, occupation, geographical region and gender. They have written the Devanagari words with black/blue pens and belongs to diverse backgrounds. Here, first 100 town/city names (collected) are similar as taken in (Shaw et al., 2008a, 2008b; Shaw and Parui, 2010; Shaw et al., 2014, 2015) . Additional

20 names of Indian States are added in the list and hence total 120 names (word-classes) are collected as database for this work. Sample of Devanagari words considered for this work along with their English version are presented in Table 3.1

Table 3.1: Devanagari words considered for this work along with their English version

Class #	Devanagari word	English version	Class #	Devanagari word	English version	Class #	Devanagari word	English version
1	आसनसोल	Asansol	41	लुधियाना	Ludhiana	81	तिरुच्चिरापल्ली	Tiruchirappalli
2	औरंगाबाद	Aurangabad	42	पोरबंदर	Porbandar	82	नेल्लोर	Nellore
3	कांकीनाड़ा	Kakinada	43	तिसतातोरसा	Teestatorsha	83	न्युफरक्का	Newfarakka
4	कपूरथला	Kapurthala	44	विराटि	Virat	84	बक्सर	Buxar
5	खजुराहो	Khajuraho	45	काकूरगाछी	Kankurgachi	85	बख्तियारपुर	Bakhtiarpur
6	ऋषिकेश	Rishikesh	46	देवघर	Deoghar	86	वर्द्धमान	Bardhaman
7	नैनीताल	Nainital	47	चित्रकूट	Chitrakoot	87	बेंगलूरु	Bengaluru
8	चौरंगी	Chowringhee	48	कोचीन	Cochin	88	मुंबई	Mumbai
9	त्रिवेणी	Triveni	49	चंदौसी	Chandausi	89	रक्सौल	Raxaul
10	वाराणसी	Varanasi	50	तंजौर	Thanjavur	90	हरिद्वार	Haridwar
11	हुगली	Hooghly	51	फिरोजाबाद	Firozabad	91	समस्तीपुर	Samastipur
12	मैसूर	Mysuru	52	पीलीभीत	Pilibhit	92	दिल्ली	Delhi
13	छपरा	Chapra	53	मुरैना	Morena	93	दार्जिलिंग	Darjeeling
14	मेरठ	Meerut	54	सिरोही	Sirohi	94	ग्वालियर	Gwalior
15	ऊटी	Ooty	55	अमृतसर	Amritsar	95	मुजफ्फरपुर	Muzaffarpur
16	झरिया	Jharia	56	ज्ञानपुर	Gyanpur	96	चेन्नैई	Chennai
17	अहमदाबाद	Ahmedabad	57	खिदीरपुर	Khidirpur	97	राजेन्द्रनगर	Rajendranagar
18	महेषतला	Maheshtala	58	गुरवायुर	Guruvayur	98	ईस्लामपुर	Islampur
19	एलौरा	Ellora	59	गोमो	Gomoh	99	भुवनेश्वर	Bhubaneswar
20	लक्षमनपुर	Laxmanpur	60	डिगबोई	Digboi	100	नवद्वीप	Nabadwip
21	इटावा	Etawah	61	चिदंबरम	Chidambaram	101	आंध्रप्रदेश	Andhrapradesh
22	राणाघाट	Ranaghat	62	झुमरीतलैया	Jhumritelaiya	102	अरुणाचल	Arunachal
23	साहिबगंज	Sahibganj	63	मयिलादुतुरै	Mayiladuthurai	103	असम	Assam
24	अंडमान	Andaman	64	मुर्तजापुर	Mirzapur	104	बिहार	Bihar
25	भरतपर	Bharatpur	65	बनमंखी	Banmankhi	105	छत्तीसगढ़	Chhattisgarh
26	हावडा	Howrah	66	मंडलाफोर्ट	Mandlafort	106	गोआ	Goa
27	जोधपुर	Jodhpur	67	चित्रदुर्ग	Chitradurga	107	गुजरात	Gujarat
28	पानागढ़	Panagarh	68	पंजाब	Punjab	108	हरियाणा	Haryana
29	विजयवाड़ा	Vijayawada	69	खुरदारोड	Khurdaroad	109	हिमाचल	Himachal
30	क्षत्रपतीनगर	Chatrapatinagar	70	भोपाल	Bhopal	110	जम्मू	Jammu
31	फरिदाबाद	Faridabad	71	भिलाई	Bhilai	111	कश्मीर	Kashmir
32	डेहरीओनसोन	Dehrionsone	72	ढोलपुर	Dholpur	112	झारखंड	Jharkhand
33	गिरिडीह	Giridih	73	ठनकपुर	Tanakpur	113	कर्नाटक	Karnataka
34	एटा	Etah	74	धौलपुर	Dholpur	114	केरल	Kerala
35	उलबेड़िया	Uluberia	75	अयोध्या	Ayodhya	115	महाराष्ट्र	Maharashtra
36	डानकुनी	Dankuni	76	उज्जैन	Ujjain	116	मणिपुर	Manipur
37	सेवड़ाफुलि	Sadafuli	77	कन्याकुमारी	Kanyakumari	117	मेघालय	Meghalaya
38	थाने	Thane	78	कृष्णनगर	Krishnanagar	118	मिज़ोरम	Mizoram
39	वैशाली	Vaishali	79	चित्तरंजन	Chittaranjan	119	नागालैंड	Nagaland
40	देहरादून	Dehradun	80	जम्मूतवी	Jammutawi	120	राजस्थान	Rajasthan

Some examples of Devanagari words depicting the shape variation in collected database of handwritten Devanagari words are given in Fig. 3.1.

Class #	Devanagari word	Writer 1	Writer 2	Writer 3	Writer 4	....	Writer N
1	आसनसोल	आसनसोल	आसनसोल	आसनसोल	आसनसोल	....	आसनसोल
2	औरंगाबाद	औरंगाबाद	औरंगाबाद	औरंगाबाद	औरंगाबाद	....	औरंगाबाद
3	कांकीनाड़ा	कांकीनाड़ा	कांकीनाड़ा	कांकीनाड़ा	कांकीनाड़ा	....	कांकीनाड़ा
4	कपूरथला	कपूरथला	कपूरथला	कपूरथला	कपूरथला	....	कपूरथला
5	खजुराहो	खजुराहो	खजुराहो	खजुराहो	खजुराहो	....	खजुराहो
⋮	⋮	⋮	⋮	⋮	⋮	....	⋮
⋮	⋮	⋮	⋮	⋮	⋮	....	⋮
116	मणिपुर	मणिपुर	मणिपुर	मणिपुर	मणिपुर	....	मणिपुर
117	मेघालय	मेघालय	मेघालय	मेघालय	मेघालय	....	मेघालय
118	मिज़ोरम	मिज़ोरम	मिज़ोरम	मिज़ोरम	मिज़ोरम	....	मिज़ोरम
119	नागालैंड	नागालैंड	नागालैंड	नागालैंड	नागालैंड	....	नागालैंड
120	राजस्थान	राजस्थान	राजस्थान	राजस्थान	राजस्थान	....	राजस्थान

Figure 3.1: Some examples of Devanagari words depicting the shape variation

3.2 DIGITIZATION

Writers have written above mentioned Devanagari words on A4-sized papers and thereafter, scanning at 300dpi has done followed by various preprocessing operations such as cropping/resizing, binarization and thinning. After scanning, words as digital images were stored in a .jpeg image format.

3.3 PRE-PROCESSING

Thereafter, pre-processing operations such as binarization using (Otsu, 1979), normalization and thinning using (Guo and Hall, 1989; Lee et al., 1994; Zhang and Suen, 1984) were carried out. Binarization converts the scanned word images into black and while pixels so as to reduce computational complexity. Whereas, normalization is used to achieve uniformity in handwritten word sizes. In this work, a uniform size of 256×64 has been considered due to horizontal writing style of Devanagari script. To minimize the text width from multiple pixels to unit-pixel, thinning operation is applied on normalized words which helps to reduce the amount of data to represent or store a word.

3.4 PERFORMANCE ANALYSIS OF THINNING ALGORITHMS FOR OFFLINE HANDWRITTEN DEVANAGARI WORDS

In this work, performance of three thinning algorithms developed by Zhang-Suen [ZSu] (Zhang and Suen, 1984), Guo-Hall [GHa] (Guo and Hall, 1989) and Lee-Kashyab-Chu [LKC] (Lee et al., 1994) have been analyzed to check their suitability to skeletonize handwritten Devanagari words in terms of various objective (reduction rate, sensitivity measurement and thinness measurement) and subjective (mean opinion score) performance metrics. For the present work, the performance of these algorithms has been tested using a handwritten Devanagari words database having 15-word classes, collected from hundreds of writers.

3.4.1 Overview of Thinning Algorithms

3.4.1.1 A Brief Overview of [ZSu] Algorithm

In 1984, Zhang and Suen proposed a thinning algorithm which shall be capable of thinning digital patterns in a efficient way (Zhang and Suen, 1984). Their algorithm removes the pixels on the object boundary by making successive passes within the entire image pixels until no more pixels can be removed. It algorithm is based on two sub-iterations. In the first sub-iteration: south-east boundary points and the north-west corner points shall be deleted. Whereas in the second sub-iteration: north-west boundary points and the south-east corner points shall be deleted while preserving the end points and pixel connectivity. In this way, after performing several iterations, only a similar look skeleton of the pattern or handwritten word shall remain with unitary thickness.

3.4.1.2 A Brief Overview of [GHa] Algorithm

Guo and Hall, (1989) algorithm is based on the removing of the border pixels at each iteration until none other pixel can be removed without shifting the connectivity. In this way, it produces a relatively thicker skeleton of digital pattern. Their algorithm is based on two sub-iteration approaches. In first approach: alternatively north and east boundary pixels, thereafter south and west boundary pixels shall be deleted. While in second approach: alternately a thinning operator shall be applied to one of two subfields. This algorithm shall result a very thin medial curves along with preserving image connectivities (Guo and Hall, 1989).

3.4.1.3 A Brief Overview of [LKC] Algorithm

Lee et al., (1994) proposed another parallel thinning algorithm which can also handle 3-D pattern/object images. It was developed for extracting both medial surfaces and axes of binary image. It was based on iterative approach. In every iterative-loop; firstly, it moves over the all pixels of a pattern and thereafter, it shall remove the undesired pixels until the pattern stops altering. In order to preserve the local-connectivity of a pattern in a better way, its every iterative-loop generally consists of two phases. In the first phase, it gathers the list of undesired pixels to be removed. Secondly, in the next phase, it rechecks the shortlisted undesired pixels of first-phase to ensure the preserved connectivity of the pattern.

3.4.2 Performance Metrics

Performance metrics serve as valuable tools to assess the quality of processed images and evaluate the effectiveness of these algorithms. In the field of research, performance metrics can generally be classified into two main categories: objective and subjective measurements. This work considers several objective performance metrics to assess the effectiveness of thinning algorithms. These metrics, include Reduction Rate (RR), Sensitivity Measurement (SM) and Thinness Measurement (TM). Mean Opinion Score (MoS) has been considered as subjective performance metrics and are briefly described in the following sub-sections (Ng et al., 1994; Chatbri and Kameyama, 2014; Goyal and Dutta, 2016).

3.4.2.1 Reduction Rate (RR)

Reduction rate is calculated on the basis of foreground pixels present in the original image and resultant skeleton of the image. Mathematically reduction rate is defined as given below in the Eq. 3.1:

$$Reduction\ Rate\ (RR)\ = \left[\frac{(fgps - fgpst)}{fgps}\right] \times 100 \qquad (3.1)$$

Where,

$$fgps\ =\ foreground\ pixels\ in\ the\ original\ image$$

$$fgpst\ =\ foreground\ pixels\ in\ the\ skeleton\ image$$

Ideally, reduction rate should be 100%. Practically, it should be high as possible.

3.4.2.2 Sensitivity Measurement (SM)

It is another qualitative metric that helps to determine wheatear the thinning algorithm has selected the best thinned image from the available scale space. The total number of cross-points present in an image can be used for the measurement of sensitivity. It is expressed by the following mathematical Eq. 3.2:

$$Sensitivity\ Measurement\ (SM)\ = \sum_{i=0}^{n}\sum_{j=0}^{m} S(P[i][j]) \qquad (3.2)$$

Where,

$$S(P[i][j]) = \begin{cases} 1, & if\ Trans(P[i][j] > 2 \\ 0, & otherwise \end{cases}$$

Lower value of SM, indicates the skeleton image contains less artifacts, redundant branches and lines caused by noise.

3.4.2.3 Thinness Measurement (TM)

The Thinness Measurement (TM) parameter measures the extent or degree to which a pattern or handwritten word present in the scanned image is thinned. Mathematically, TM can be calculated as given in the following Eq. 3.3:

$$Thinness\ Measurement\ (TM) = \left(1 - \frac{TM_1}{TM_2}\right) \qquad (3.3)$$

Where,

$$TM_1 = \sum_{i=0}^{n}\sum_{j=0}^{m} triangle\_count(P[i][j])$$

$$TM_{2} = 4 \times [\max(\text{height}, \text{width}) - 1]^2 = 4 \times [\max(\text{m}, \text{n}) - 1]^2$$

Its value range is of $[0, 1]$. $TM = 1$ indicates that pattern or handwritten word is completely unit pixel wide.

3.4.2.4 Mean Opinion Score (MoS)

The performance analysis of thinning algorithms is also definitive measure of the quality of the skeleton image. The thinned image quality may be specified by Mean Opinion Score (MoS), which is the result of the perception based subjective evaluation. The 5-level grading scores of MoS (i.e. 5-excellent, 4-good, 3-acceptable, 2-poor quality and 1-unaccepable) have been considered for this work too. In the following sections the simulation results have been represented for various thinning algorithms based upon above performance metrics.

3.4.3 Performance Analysis and Discussion

Performance analysis of various thinning algorithms has been carried out on a common set of HDW samples (15-word classes) collected from hundreds of writers so as to check the suitability of the algorithms for the same. The 15 word-classes taken for this work are namely "आसनसोल" (Asansol), "औरंगाबाद" (Aurangabad), "कांकीनाड़ा" (Kakinada), "कपूरथला" (Kapurthala), "खजुराहो" (Khajuraho), "ऋषिकेश" (Rishikesh), "नैनीताल" (Nainital), "चौरंगी" (Chowringhee), "त्रिवेणी" (Triveni), "वाराणसी" (Varanasi), "हुगली" (Hooghly), "मैसूर" (Mysuru), "छपरा" (Chapra), "मेरठ" (Meerut) and "ऊटी" (Ooty). For each word, the quality of resultant thinned word image is evaluated using objective as well as subjective performance metrics. Firstly, each HDW samples were collected using A-4 sized paper, then scanned, normalized and thereafter, converted into binary image using Ostu's threshold selection method. Results for two handwritten word-classes namely "कांकीनाड़ा" (Kakinada) and "ऊटी" (Ooty) are depicted before and after applying Ostu's threshold (Otsu, 1979) along with their histograms in the Fig. 3.2 and Fig. 3.3 respectively.

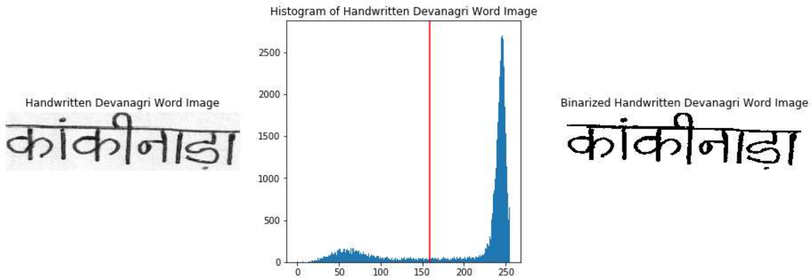

Figure 3.2: Handwritten Devanagari word "कांकीनाड़ा" (Kakinada) before and after applying Ostu's method along with its histogram

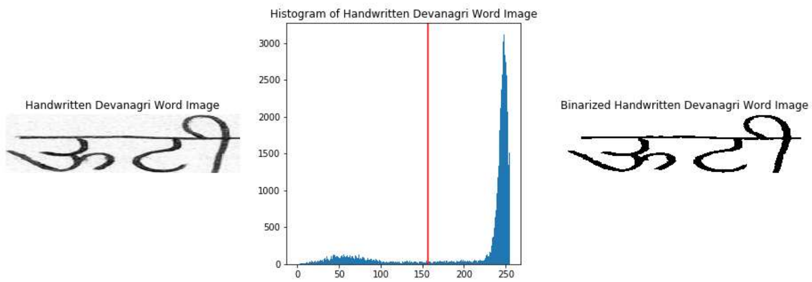

Figure 3.3: Handwritten Devanagari word "ऊटी" (Ooty) before and after applying Ostu's method along with its histogram

Corresponding binarized HDWs are further processed and various thinning algorithms are applied on them. Before applying various thinning algorithms to Ostu's thresholded handwritten word samples (binarized image), the each samples of binary HDW are inverted which shall turn white pixels into black and vice-versa. After that thinning algorithms namely [ZSu], [GHa] and [LKC] have been applied/implemented on a common set of handwritten Devanagari word samples (15 word-classes) using Scikit-learn library available in Python. Pictorial representation of two HDW samples (2 word-classes) namely "कांकीनाड़ा" (Kakinada) and "ऊटी" (Ooty) are represented in the following Figs. 3.4 and 3.5.

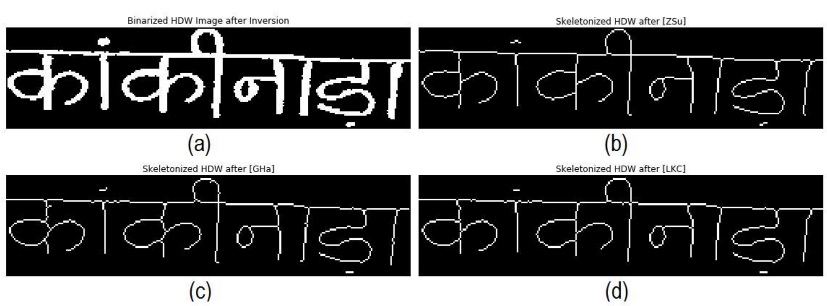

Figure 3.4: Handwritten Devanagari word "कांकीनाड़ा"(Kakinada) and its resultant skeleton
(a) After Ostu's threshold and inversion **(b)** After Zhang and Suen (1984)
(c) After Guo and Hall (1989) and **(d)** After Lee et al. (1994)

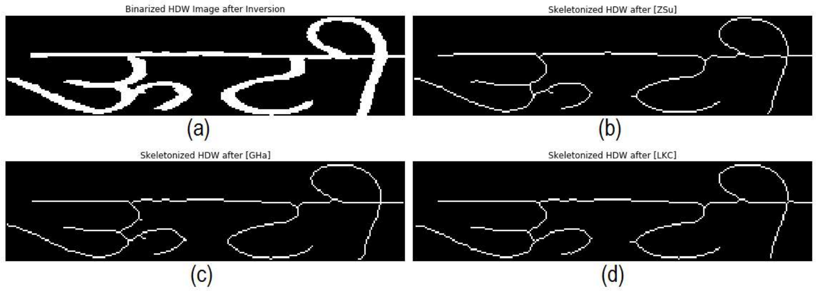

Figure 3.5: Handwritten Devanagari word "ऊटी"(Ooty) and its resultant skeleton **(a)** After Ostu's threshold and inversion **(b)** After Zhang and Suen (1984) **(c)** After Guo and Hall (1989) and **(d)** After Lee et al. (1994)

To check the suitability of these algorithms to thin or skeleton offline HDW, various available objective performance metrics namely reduction rate, sensitivity measurement and thinness measurement are calculated for various HDW samples of 15 different word-classes. The average measurements of these performance metrics are presented in the following Tables 1 to 3 respectively.

Table 3.2: Reduction Rate (RR)

HDWs	**HDWs [English Version]**	**Zhang-Sue [ZSu]**	**Guo-Hall [GHa]**	**Lee-Kashyab-Chu [LKC]**
आसनसोल	Asansol	72.21705426	72.80620155	73.30232558
औरंगाबाद	Aurangabad	73.95287958	74.05104712	74.67277487
कांकीनाड़ा	Kakinada	73.01197421	73.28830212	73.62603623
कपूरथला	Kapurthala	76.60914818	76.93782525	77.18433306
खजुराहो	Khajuraho	71.6117851	71.95840555	72.6169844
ऋषिकेश	Rishikesh	76.45631068	76.69902913	77.21143474
नैनीताल	Nainital	77.32476962	77.71572187	78.02289863
चौरंगी	Chowringhee	77.30870712	77.54324245	78.39343301
त्रिवेणी	Triveni	79.24880128	79.43526905	79.70165157
वाराणसी	Varanasi	76.66281087	76.83632157	77.06766917
हुगली	Hooghly	69.76058932	70.68139963	71.12338858
मैसूर	Mysuru	74.53838678	74.99190152	75.5749919
छपरा	Chapra	81.56970913	82.02106319	81.87061184
मेरठ	Meerut	79.04389657	79.07396272	79.70535177
ऊटी	Ooty	75.73721538	75.47592385	75.73721538
Average Reduction Rate		**75.67026921**	**75.96770777**	**76.38740672**

Table 1. shows that [LKC] algorithm has achieved higher average reduction rate as compared with [ZSu] and [GHa] algorithms for HDWs. Moreover, both [ZSu] and [GHa] algorithms resulted nearby figures in terms of reduction rate. Graphical representation of the same has been depicted in the following Fig. 3.6. This shows ability of thinning algorithms to reduce the foreground pixels in original pattern/word.

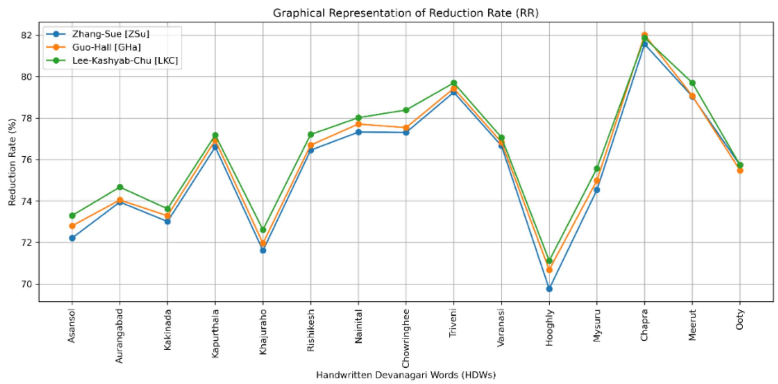

Figure 3.6: Graphical representation of Reduction Rate (RR)

Table 3.3 shows that both [ZSu] and [GHa] algorithms has achieved lower average sensitivity measurement as compared with [LKC] algorithm for HDWs.

Table 3.3: Sensitivity Measurement (SM)

HDWs	HDWs [English Version]	Zhang-Sue [ZSu]	Guo-Hall [GHa]	Lee-Kashyab-Chu [LKC]
आसनसोल	Asansol	0.441860465	0.453023256	0.463875969
औरंगाबाद	Aurangabad	0.478075916	0.480039267	0.492473822
कांकीनाड़ा	Kakinada	0.45901136	0.464844949	0.471292601
कपूरथला	Kapurthala	0.532456861	0.5393043	0.543960559
खजुराहो	Khajuraho	0.432235702	0.43847487	0.451299827
ऋषिकेश	Rishikesh	0.526968716	0.531823085	0.54180151
नैनीताल	Nainital	0.544540631	0.552080424	0.558503211
चौरंगी	Chowringhee	0.54353562	0.548226327	0.565523307
त्रिवेणी	Triveni	0.581513053	0.585242408	0.590570059
वाराणसी	Varanasi	0.530075188	0.533545402	0.537883169
हुगली	Hooghly	0.39558011	0.413627993	0.422099448
मैसूर	Mysuru	0.488824101	0.498218335	0.509556203
छपरा	Chapra	0.630892678	0.639919759	0.636910732
मेरठ	Meerut	0.579374624	0.580276609	0.592603728
ऊटी	Ooty	0.51399776	0.509145203	0.514371034
Average Sensitivity Measurement		**0.511929519**	**0.517852813**	**0.526181679**

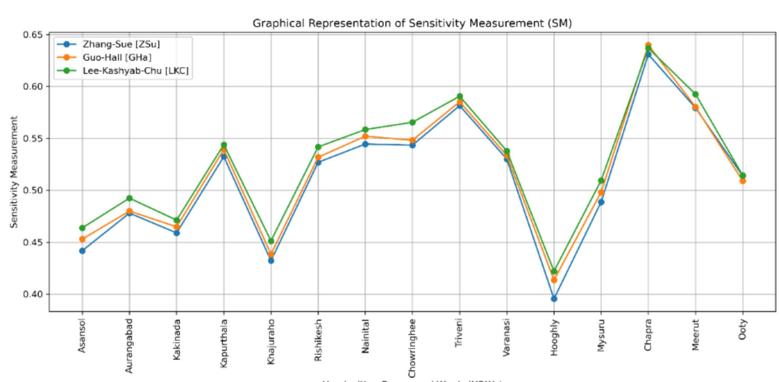

Figure 3.7: Graphical representation of Sensitivity Measurement (SM)

Graphical Representation of Sensitivity Measurement (SM) has been depicted in the Fig. 3.7. Therefore, it can be summarized that resultant thinned image by [LKC] algorithm may contain some artifacts, redundant branches and lines caused by noise.

Table 3.4 shows that [LKC] and [GHa] algorithms has achieved higher average thinness measurement as compared with [ZSu] algorithm for HDWs.

Table 3.4: Thinness Measurement (TM)

HDWs	HDWs [English Version]	Zhang-Sue [ZSu]	Guo-Hall [GHa]	Lee-Kashyab-Chu [LKC]
आसनसोल	Asansol	0.983855945	0.991927973	0.993790748
औरंगाबाद	Aurangabad	0.982651391	0.988543372	0.993126023
कांकीनाड़ा	Kakinada	0.986797667	0.992938287	0.996315628
कपूरथला	Kapurthala	0.988209487	0.993693447	0.996709624
खजुराहो	Khajuraho	0.984049931	0.990984743	0.995839112
ऋषिकेश	Rishikesh	0.984889369	0.989206692	0.995412844
नैनीताल	Nainital	0.989667691	0.990784697	0.996648981
चौरंगी	Chowringhee	0.986788021	0.991779213	0.997651204
त्रिवेणी	Triveni	0.985348961	0.991742142	0.99786894
वाराणसी	Varanasi	0.986982933	0.990743419	0.997107318
हुगली	Hooghly	0.983539095	0.99251777	0.994014216
मैसूर	Mysuru	0.985026042	0.991861979	0.994791667
छपरा	Chapra	0.990966123	0.995734003	0.99749059
मेरठ	Meerut	0.990977444	0.993082707	0.99518797
ऊटी	Ooty	0.99252895	0.996264475	0.99850579
Average Thinness Measurement		**0.986818603**	**0.992120328**	**0.99603071**

Graphical representation of the thinning measurement for various thinning algorithms has been depicted in the following Fig. 3.8. This shows ability of thinning algorithms to produce unit-pixel wide skeleton of original pattern/word.

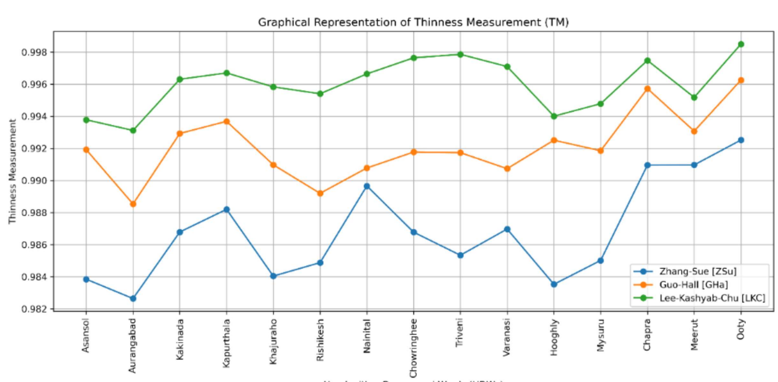

Figure 3.8: Graphical representation of Thinness Measurement (TM)

The analysis indicates that the Lee-Kashyab-Chu [LKC] thinning algorithm exhibits slightly superior performance in terms of reduction rate and sensitivity measurement compared to the Zhang-Sue [ZSu] and Guo-Hall [GHa] algorithms when applied to the collected handwritten Devanagari word (HDW) samples. Regarding subjective performance metrics, specifically the Mean Opinion Score (MoS), the skeletons generated by the LKC algorithm achieved a higher MoS grade-4 (good) in comparison to the other two thinning algorithms. Additionally, both the Guo-Hall [GHa] and Zhang-Suen [ZSu] algorithms obtained an MoS grading score-3 (acceptable) for the considered HDW database in this study.

3.5 CHAPTER SUMMARY

Based on the findings of this study, the Lee-Kashyab-Chu [LKC] algorithm outperformed the other two algorithms in terms of Reduction Rate (RR), Thinness Measurement (TM), and Mean Opinion Score (MoS). However, it is worth noting that this algorithm achieved a slightly higher value of Sensitivity Measurement (SM) compared to the other mentioned algorithms. This suggests that the thinned images generated by the [LKC] algorithm may contain some artifacts, redundant branches, and lines caused by noise, in comparison to the other algorithms considered in this study. Consequently, the thinning scheme could serve as a valuable pre-processing method for handwritten Devanagari word recognition.