



CS-299
Innovative Design Laboratory

Hindi OCR Using Neural Networks

Abhishek Nautiyal
(1601CS02)
Piyush Singh
(1601CS30)

ABSTRACT

English Character Recognition has been extensively studied in the last half century and progressed to a level, sufficient to produce technology driven applications. But same is not the case for Indian languages which are complicated in terms of structure and computations. As there is no separation between the characters of texts written in Hindi as there is in English it further complicates the segmentation process, creating a major problem when designing an effective character segmentation technique.

Rapidly growing computational power may enable the implementation of Hindi OCR methodologies. Digital document processing is gaining popularity for application to office and library automation, banks and postal services, publishing houses and communication technology, to nursing homes and hospitals.

Hindi is the most widely spoken language in India, with more than 300 million speakers, should be given special attention so that document retrieval and analysis of rich ancient and modern Indian literature can be effectively done.

AIMS, OBJECTIVES & SIGNIFICANCE

Converting the image of a machine printed Hindi document into a editable format. This project holds great significance since it aims to assist in easing the conversion from physical to electronic type. Such capacity holds significant credibility and its advantages are limitless. This converts handwritten or printed symbols from simple pictures to helpful information that may be utilized in computers. These handwritten or printed documents do not stay in a large pile of pages in the workplace, instead they are now turned into digital information that can be easily interpreted by the computers. This also makes the process of searching these documents easy. Information is turned into digital format without any individual having to do the tedious work himself. Digital document processing has large applications in offices, banking services, communication technology.

Therefore, it becomes obvious that OCR platforms are incredibly useful for such functions and could be manipulated more to be employed in various other tasks of daily life that can be personal as well as commercial in nature

RELATED WORK

Most OCR techniques use split words into smaller units, though implementations vary on the level of segmentation. While some approaches use character as classification units others segment a character into components before classification. When characters used as classification units, horizontal and vertical profile of each word is examined to remove the header line. In Hindi words, the header line is very well-defined and is the basis for modelling classification techniques.

Some Popular reference works are cited below :

- Kailash S. Sharma, A. R. Karwankar, Dr. A.S. Bhalchandra, "Devnagari Character Recognition Using Self Organizing Maps" ICCCT'10
- H. Ma and D. Doermann, "Adaptive Hindi OCR using generalized Hausdorff image comparison," ACM Trans. Asian Lang. Inf. Process. vol. 2, no. 3, pp. 193–218, 2003.
- R.M.K. Sinha, and Veena Bansal, "On Automating trainer for construction of prototypes for Devnagari text recognition", Technical report TRCS-95-232, IIT Kanpur, India 1995.
- U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 3, pp. 444–457, Mar. 2009.
- U. Pal and B. B. Chaudhuri, "Indian script character recognition: A survey," Pattern Recognit., vol. 37, pp. 1887–1899, 2004.
- R.M.K. Sinha, and Veena Bansal, "On Devanagari documentation processing", IEEE International Conference on Systems, Man and Cybernetics, Vancouver, Canada 1995.
- Veena Bansal, R.M.K. Sinha, "On How to Describe Shapes of Devanagari Characters and Use Them for Recognition," icdar, pp.410, Fifth International Conference on Document Analysis and Recognition (ICDAR'99), 1999
- Veena Bansal & R.M.K. Sinha, "Segmentation of Touching Characters In Devanagari", <http://www.iitk.ac.in/ime/veena/PAPERS/stwo.pdf>
- M. Babu Rao, Dr. B. Prabhakara Rao, Dr. A. Govardhan, "Content Based Image Retrieval using Dominant Color and Texture features" (IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 2, February 2011
- R. Jayadevan, Satish R. Kolhe, Pradeep M. Patil, and Umapada Pal "Offline Recognition of Devanagari Script: A Survey", IEEE Transactions on Systems, Man, and Cybernetics—part C: Applications and Reviews, vol. 41, no. 6, November 2011.

NOVELTY

As we know that recognition of character is not an easy task. Due to various font sizes and writing style it is difficult to recognize the character. Also in Devnagari script, many characters have similar shape, which creates trouble in recognition.

While in English language words are essentially isolated alphabets printed in close proximity, on the other hand in Hindi a large number of characters can be formed using existing characters and thus increasing the difficult several folds.

स + त = स्त	sta
ष + ठ = ष्ठ	ṣṭa
क + ष = क्ष	kṣa
ह + ण = ह्र	hṇa
म + प + र = म्प्र	mpa

Also unlike alphabets in English, Hindi characters consists of three different layers listed as follows :

Ascender →
Core →
Descender →

Shirorekha

पुनीत

पु नी त Character or Orthographic syllable

प ७ न ी त Alphabet or glyph

प ७ न २ १ त Component

These additional attributes of the hindi language and the presence of more than 50 unique characters that can conjoin to form further more new characters makes this project exciting and hard at the same time!

The DATASET

Table 1 Class space of our system (components)

।	च	थ	र	क	८
अ	छ	द	ल	रु	फ
इ	ज	ध	ळ	ॢ	ॠ
उ	झ	न	व	ॣ	ॡ
ऊ	ट	प	श	ज	ॢ
ए	ठ	फ	ष	झ	ॣ
क	ड	ब	स	।	।
ख	ढ	भ	ह	॥	॥
ग	ण	म	क्ष	॥	॥
घ	त	य	त्र	॥	॥

Consonants and vowels

Ascenders

Half-consonants

Descenders

WORK OUTLINE & PLAN OF WORK

The main steps involved in Character Recognition are as follows :

- Text Digitization
- Gray Tone to Two Tone Conversion
- Noise clearing
- Text Block Identification
- Skew Correction
- Line and Word Detection
- Character Segmentation

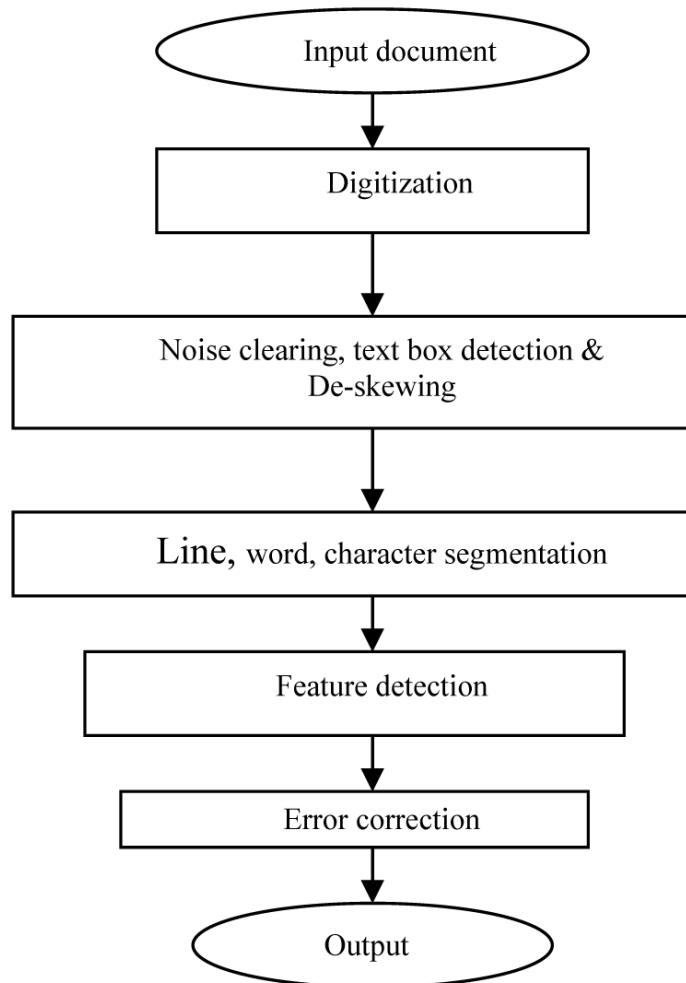


IMAGE PREPROCESSING

Data in a paper document are usually captured by optical scanning and stored in a file of picture elements, called pixels. These pixels may have values: OFF (0) or ON (1) for binary images, 0– 255 for gray-scale images, and 3 channels of 0–255 colour values for colour images. This collected raw data must be further analyzed to get useful information.

Such processing includes the following:

1. Thresholding :

A grayscale or colour image is reduced to a binary image.

2. Noise reduction :

The noise, introduced by the optical scanning device or the writing instrument, causes disconnected line segments, bumps and gaps in lines, filled loops etc. The distortion including local variations, rounding of corners, dilation and erosion, is also a problem. Prior to the character recognition, it is necessary to eliminate these imperfections.

3. Skew Detection and Correction :

Documents may originally be skewed or skewness may introduce in document scanning process. This effect is unintentional in many real cases, and it should be eliminated because it dramatically reduces the accuracy of the subsequent processes, such as segmentation and classification. Skewed lines are made horizontal by calculating skew angle and making proper correction in the raw image.

4. Edge-Detection :

The boundary detection of image is done to enable easier subsequent detection of pertinent features and objects of interest by using the canny edge detector.



SEGMENTATION

It is one the most important process that decides the success of character recognition technique. It is used to decompose an image of a sequence of characters into sub images of individual symbols by segmenting lines and words. Hindi words can further be splitted to individual character for classification and recognition by removing Shirorekha (header line). The Shirorekha is removed by applying line detection using Hough Transform method. Various vowel modifiers can be separated for structural feature extractions.

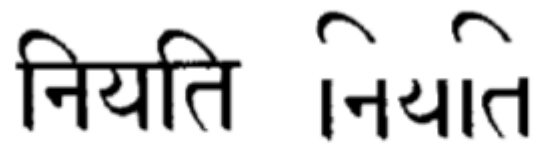
The image shows the Hindi word 'नियति' (Niyati) in a black, sans-serif font. The word is split into two parts, 'नियति' and 'नियति', with a significant gap between them. This illustrates the removal of the Shirorekha (header line) from the original word, which is a step in character segmentation for Hindi text recognition.

Figure 1. Shirorekha removal for a Hindi word

FEATURE DETECTION AND CLASSIFICATION

Feature extraction and selection can be defined as extracting the most representative information from the raw data, which minimizes the within class pattern variability while enhancing the between class pattern variability. For this purpose, a set of features are extracted for each class that helps distinguish it from other classes, while remaining invariant to characteristic differences within the class.

In this case, we will be going to apply the HOG image de-scriptor and a Linear Support Vector Machine (SVM) to learn the representation of characters. For this system, we used python, openCV and sklearn to run classification and read the dataset.

The sample of the dataset consists of 30000 data points, each with a feature vector of length 784, corresponding to the 28×28 grayscale pixel intensities of the image.

Support Vector Machine (SVM)

SVM are a set of supervised learning methods used for classification, regression and outliers detection. In SVM, each data item will be plot as a point in n-dimensional space (n = number of feature) with the value of each feature being value of a particular coordinate. The classification is done by finding the hyper-plane that differentiate the two (2) classes.

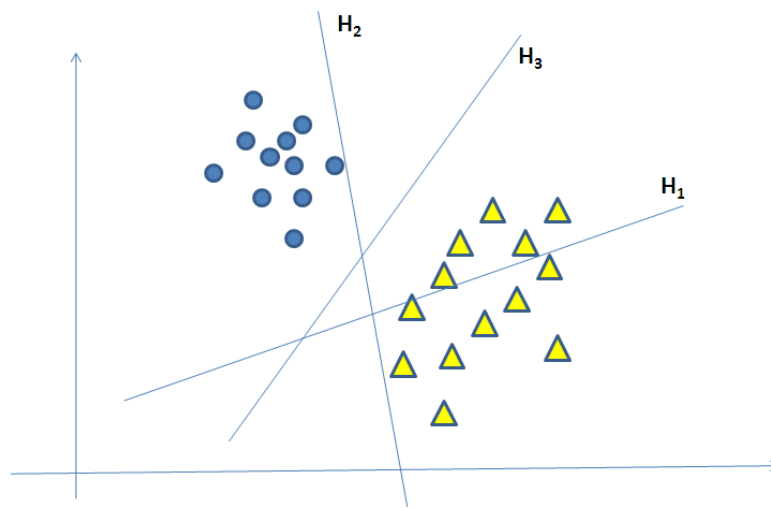


FIGURE: Separation hyperplanes. H1 does not separate the two classes; H2 separates but with a very tiny margin between the classes and H3 separates the two classes with much better margin than H2.

The advantage of SVM classifier over other classifiers can be explained as follows. An Indian language OCR system generally has large number of classes and high dimensional feature vectors. Variability of characters is also very high at each occurrence. SVMs are well suited for such problems since they have excellent generalization capability. They always result in a global solution of the problem.

The SVM in scikit-learn support both dense `numpy.ndarray` and convertible to that by `numpy.asarray`) and sparse (any `scipy.sparse`) sample vectors as input. In scikit-learn have three (3) classes that capable of performing multi-class classification on a dataset which is SVC, NuSVC and LinearSVC. In this system we will use LinearSVC class to perform the classification on our dataset. LinearSVC or Linear Support Vector Classification that use linear kernel and implemented in terms of liblinear that has

more flexibility in the choice of penalties and loss functions and should scale better large numbers of samples.

In this classification, we will have two (2) file which is `classify.py` for classification and `predict.py` for testing the classification. In the `generateClassifier.py` we will perform three (3) step such as:

- Calculate the HOG features for each sample in the database.
- Train a multi-class linear SVM with the HOG features of each sample along with the corresponding label.
- Save classifier in a file.

The dataset images of the digits will be save in a numpy array and corresponding labels. Next we will calculate the HOG features for each images and save them in another numpy array.

Computing the HOG descriptor is handled by the `hog` method of the feature sub-package of `scikit-image`. We pass in the number of orientations, pixels per cell, cells per block, and whether or not square-root transformation should be applied to the image prior to computing the HOG descriptor.

Then we will create Linear SVM object and perform the training for the dataset then we will save the classifier in a file.

Now to classify a test image, we compute the HOG feature vector of the thresholded ROI(Region of Interest) by calling the `describe` method of the HOG descriptor.

The HOG feature vector is then fed into the `LinearSVC`'s `predict` method which classifies which character the ROI is, based on the HOG feature vector.

RESULTS & OBSERVATIONS

The results reported here are character level while in real-life applications of the system we will be comparing with recognized sentences and at paragraph levels.

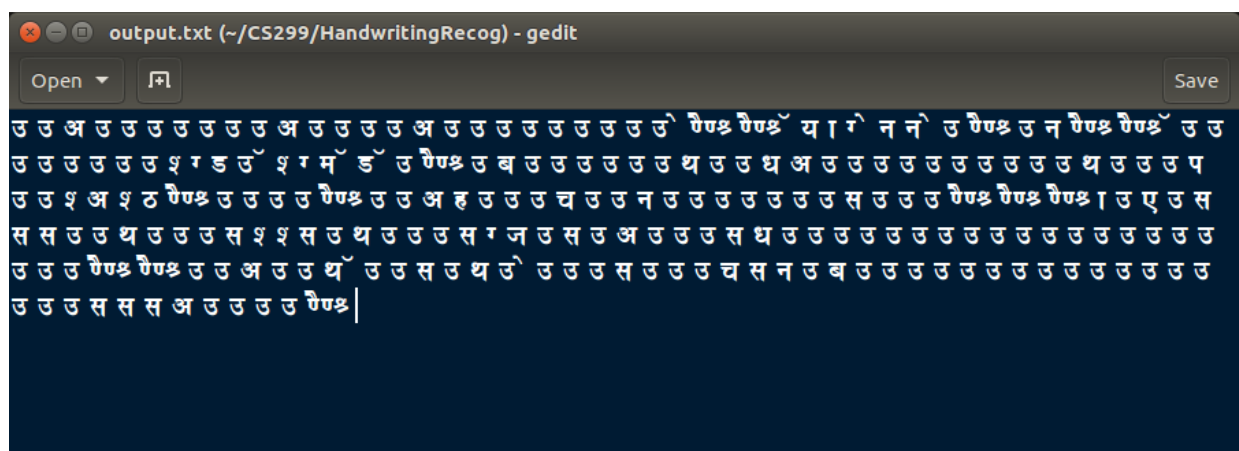
Accuracy rate is calculated by first calculating the total sum of the given letter and then finding the percentage of given letter where it's false. As you see, some letters have a very low accuracy

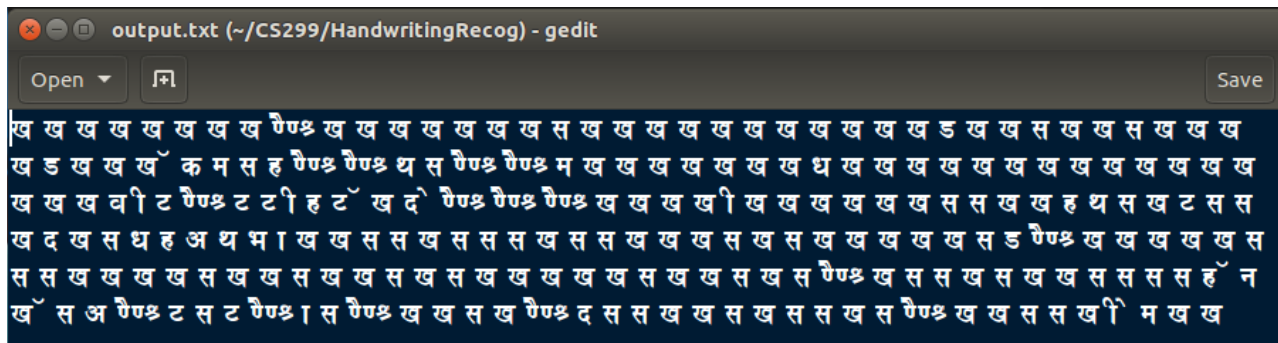
whereas the accuracy of other letters is satisfactory.

After training the given letters with more data sets, results are improved.

However, there has been a small decline in some of the letters. The main reason for the decline is because of the increasing similarity of 2 letters and this problem can be solved by writing them in different styles. It's still not guaranteed that the accuracy rate will reach 100% but it will slightly increase if the image is well processed by the system.

ਤ
ਤ
ਤ
ਤ
ਤ
ਤ
ਤ ਤ



[illegible]

In this example the programs gets confused between two similar characters i.e., and as a result there is a good number of times that it gives that character as an output.

While in other cases due to low number of training images the accuracy is low. For such cases we need to drastically improve the number and quality of the training dataset, one such example is:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 104

मानरचमग्गाचाधानधरारसजरजनअघनधझरैण्वराजहारहह
रहकहचाागरगरसगरगरगरगरहहगरगरगरआगरमगरसर
सहरहगरगरगरसरअजरगरगरहाहकरभहगरगरगरकरगरहगरहगर

CONCLUSION

The experimental results illustrate that the Machine Learning concept can be applied successfully to solve the Hindi optical character recognition problem. There are many variations of factors that affect the performance of the developed Hindi OCR software.

However, other kinds of preprocessing and neural network models may be tested for a better recognition rate in the future. The character segmentation method can be improved to handle larger variety of touching characters that occur often in images obtained from inferior-quality documents. The test set used in this experiment is of 77 characters of different types of fonts. This can be tested for a greater number of fonts. The toughest phase in the experiment is getting a good set of characters for classification. This highlights the need for generation of a large ground-truthed set of characters of various resolutions so that more research can be performed for recognition of languages from Indian subcontinent.

Additional work can also be completed in order to improve the outcomes and solve the problems that were confronted with by the training of a bigger set of images.

The system cannot segment certain type of characters that are formed by adding small modifiers to existing consonant or formed by addition of 3 or more characters, this is because they comprise a small percentage of the of the language and hence sufficient sample cannot be used.

FUTURE WORK

The other future enhancement that can be incorporated in the work is to use a dictionary of words to correct the output. Certainly this will improve the performance. Further speech synthesizer can be integrated with the OCR with the aim of making a system for reading aids to the blind. We can also implement the neural network method for classifying hand-written texts.

In hand-written documents, the fragmentation of characters and the variation in shape of characters are considerably greater compared to printed documents. The higher levels can be used to provide clues for a hypothesization system, which learns from the text it recognizes. We have not dealt with punctuation marks and numerals in this work. The set of the punctuation marks and the numerals need special treatment right from the point of preliminary segmentation of words into characters and symbols. For Devanagari script, the header line is removed from the character before an attempt is made to classify it. However, a numeral or punctuation mark cannot be dealt with the same manner. There is no header line to be removed, even though some such marks may have a horizontal line which resembles the header line. We implemented the work for the scripts with only Hindi characters. However, we can extend it to classify the document with characters of more than one script.