Automatic Handwritten Devanagari Text Generation in Marathi Styles using Ant Miner Algorithm

Vajid Khan, Yogesh Kumar Sharma

***Abstract:** The Devanagari scripts forms the backbone of the writing system of several Indian languages includes Hindi, Sanskrit and Marathi. With the increased demand, exploration and globalization of digital Devanagari documents, different printed and handwritten document recognition techniques have involved since last two decades. In literature many methods of Devanagari script recognition have been used but it is not able to attain the best results in recognition. Hence, in this paper is proposed Ant Miner Algorithm (AMA) for recognition and text generation of handwritten Devanagari Marathi Scripts. The proposed method recognition process is working with the four different stages such as pre-processing, segmentation, feature extraction and recognition with text generation. The first stage pre-processing is consists of skew correction, noise removal and binarization. The second stage is segmentation that contains the line segmentation, word segmentation and character segmentation. The third stage is feature extraction method it contains four methods such as Scale Invariant Feature Transform (SIFT), Linear Discriminant Analysis (LDA), Discrete Cosine Transform (DCT) and Local Binary Pattern (LBP). The final stage is recognition and text generation with attain with the help of AMA algorithm. It works based on the two phases such as training and testing phase. The proposed method is implemented in the python platform and it compared with the Artificial Neural Network (ANN) and K-Nearest Neighbours (KNN). The performance of the proposed method is analysed with statistical measurements of accuracy, precision and recall*.

***Keywords** :Devanagari script, Marathi, AMA, feature extraction, pre-processing, segmentation.*

I. INTRODUCTION

In India a lot of official languages are written in Devanagari script which includes Sanskrit, Sindhi, Konkani, Hindi, Nepali and Marathi and it is being used by lot of peoples around the world. Other languages such as Bengali, Gujarati and Punjabi were written using similar script as that of Devanagari script [1, 3]. Because of more significance many researches has contributed more efforts in this Devanagari script. This script consists of 34 consonants with 14 modifier vowels and 13 vowels and it is written from left to right [4, 5, and 6]. The conception of upper or lower case is not present in this script.

In compound character the one half characters is linked to full character in order to create special character. Another special characteristic of this script is the existence of horizontal line on top of every character. The Devanagari script is not that must develop as that of English because of many reasons [8].The reasons behind for undevelopement which include existence of fused and touching characters, variability in writing style, presence of multiple forms of writing the same character, lack of ground truth dataset and standard benchmarking and shortage of complexity of grammatical pattern of the sentences [8, 10].

The handwritten character recognition seems to be a difficult task for Devanagari script in pattern recognition, artificial intelligence and image processing. The mechanism behind character recognition is to translate any type of written document file into text format which is editable. Many authors valuable information can be easily written and shared to the readers in digital format rather than the paper format. This digital format helps to have easy sharing of information [11].For converting into digital the character recognition plays a vital role. The character recognition is of two types namely, Optical character recognition (OCR) is also called as off-line character recognition. The handwritten, printed text or type written can be transformed to digital format. Intelligent character recognition is otherwise called as online character recognition. Based on direction of motion during writing the characters are recognized. This way of character recognition is mainly used on cell, touchpad; touch screen, etc [12]. The accuracy in recognition also gets altered because of the above mentioned reasons. So, for attaining better accuracy the recognition process must be done effectively [14].

Many Authors are developed different methods for recognition of the Devanagari script. The existing method is processing for recognize the Devanagari script which are discussed with notable performances. Generally, the recognition process mainly consist of three steps pre-processing, extraction of feature and finally classification. For character recognition different algorithm were being developed with different advances which include neural network algorithm (NNA), pattern matching algorithm (PMA), structural algorithm (SA), support vector machine algorithm (SVMA), statistical algorithm, hidden markov model (HMM) and template matching algorithm(TMA). In template matching algorithm only the typewritten characters can be recognized. But the other algorithm like neural network algorithm (NNA), structural algorithm (SA), and support vector machine (SVM) can recognize both handwritten and type written [15].

Revised Manuscript Received on December 30, 2019.
***** Correspondence Author

Mr. Vajid N Khan*, Research Scholar, Department of Computer science and engineering ,Shree JJT University, Rajsthan , India.
Dr. Yogesh Kumar Sharma, Head of Department of computer science and engineering, Shree JJT University, Rajsthan, India.

Every algorithm contains both advantages and disadvantages. To overcome the drawbacks in the previous methods, the proposed method will be developed and designed for recognition of the Devanagari scripts. In this paper, proposed method is introduced to recognition of the Devanagari script and overcome the previous methods. The main contribution of the paper is mentioned follows,

Contribution and organization of the study

The proposed method has some main contribution to attain the recognition of the Devanagari scripts.

- ✓ To develop AMA for recognizing and text generation the devanagiri scripts.
- ✓ The proposed method can be used for both handwritten and printed documents.
- ✓ The recognition and text generation of the Devanagari script can be achieved with the different process such as pre-processing, segmentation, feature extraction and recognition with text generation.
- ✓ The pre-processing consists of noise removal, binarization and skew correction; segmentation consists of word segmentation, character segmentation and line segmentation; Feature extraction consists of LBP, SIFT, LDA and DCT; Recognition and text generation is achieved with the help of the AMA algorithm.
- ✓ The proposed method is implemented in python platform and it is compared with the existing methods such as ANN and KNN.

The remaining part of the paper is organized by follows; in section 2 review papers were studied and some of them were discussed in detail. The drawbacks in the existing method are also mentioned in addition to it. The section 3 comprises the proposed part and a detail description is given in it. In section 4 the results that are simulated on implementation is included. Further the results are analyzed and the performances are compared with the existing method. In section 5 the proposed method is concluded with highlighting the significance. The section 6 includes all the references that were studied and analyzed in order to propose this work.

II. REVIEW CRITERIA

Many researchers and scholars have used lots of methodologies and technologies for recognition of printed and handwritten characters of different languages. Some of the author’s works are reviewed below,

Milind bhaerola et al., [21] have examined the OCR for scanned image of handwritten or printed characters. For accurate recognition the developed system separates characteristics on the basic of only gradient image. In developed method the combination of classifiers namely SVM and quadratic classifiers were used for recognition. The characteristic like histogram of gradient (HOG), angle and strength was used for analyzing. The angle and the strength feature are found using the Gaussian filter. Further these features were joined with HOG features. These were applied to the classifier combination in order to achieve extreme accuracy of 95.81/% with the help of three fold cross validation.

G.B kshirsagar et al., [22] have described machine learning method using P300 speller as input for Devanagari script. On evaluating the performance this method seems to have certain disadvantage like less information transfer rate (ITR). This takes place due to huge size of display needed which was subjected to task difficulty, crowding effect, fatigue ,adjacency and needed many number of trials for recognition. So, deep learning algorithms have been used for detecting P300. The two deep learning algorithms that were used are deep convolution neural network (DCNN) and stacked auto encoder (SAE). From the experimental result it was found that the DCNN can detect with 88.23% accuracy within three trials. It also gives ITR with 20.58 bits per minutes which is considerably senior than the already existing techniques.

Ayan kumar bhunia et al., [23] have presented cross language framework for recognition of character in Indic script and it was also used for spotting text. In place of insufficient training data the large number of dataset was broken by the cross language framework for training purpose for recognition and spotting text in the scripts. The Indic scripts were written in three major zones upper, middle and lower. The performance of cross language framework was evaluated using these three zones. In upper and lower zone HMM classifier was used and in middle zone SVM classifier was used and it also depends on the analogues between target and source scripts. This framework was tested in three Indic scripts such as Devanagari, Bangala and Gurumukhi and the results were analyzed and reported.

Prathiba singh et al., [24] have developed neural networks for the purpose of recognition of handwritten devanagiri script. The two types of neural network were used. One was fully connected feed forward neural network and the other one was the deep convolution neural network. Deep learning was biological technique and it was based on human brain. The part of human brain called neocortex has the layered architecture. The CNN (convolutional neural network) does not need any complicated pre-processing and feature extraction algorithms. This was the main advantage of using CNN. The image pixels were the input for these two networks. Using character benchmarking dataset improved recognition accuracy of 96.8% was achieved.

Parule sahare et al., [25] have designed SVM and k-NN (k-Nearest Neighbour) classifier for character segmentation and recognition of documental images of Devanagari and Latin script. Most of the documents have the problem of local skews, intermixed text and low print quality. In the designed algorithm primary segmentation was achieved through structural property of the characters and the graph distance theory was used for joined or overlapped character. The final segmentation is analyzed using SVM classifier. For recognizing the character k-NN classifier was used. The results are analysed using different datasets and the benchmarking dataset was found as the effective one. From the experimental result it was found that the segmentation and recognition of 98.9% was achieved. The performance of the developed method was found to better on comparing with the other existing methods.

The recognition of characters in different languages is required in order to transfer the paper format into digital format for easily sharing the valuable information of authors. With the aid of digital format the information can be easily formatted, edited and quickly shared. A large number of methods are used for recognition of characters in different languages. Handwriting dataset, HMM, digital image with primitive characteristic information, machine learning algorithm etc were used for recognition of characters. But every method contains certain limitations. In case of dataset there are limited dataset for handwritten character recognition. So, there is lack in dataset for handwritten recognition. On using HMM only 90% of accuracy in recognition can be achieved. In primitive feature recognition the character classification takes a longer time. The optical character recognition can be only done on scanned image and it is not suited for gradient image. The usual machine learning algorithm has the limitation like ITR. In [21] author designed OCR for handwritten and printed documents by means of scanned images. In this case for achieving accuracy the characters are separated based on only gradient image and combination of classifier must be used. In [22] author designed machine learning algorithm for character recognition. On using algorithm there is lack in performance due to low ITR. In [23] the author developed cross language framework for character recognition and spotting text for Indic scripts. Based on only three zones this framework can recognize the character and separate classifiers are necessary for every zone. In [24] author developed two types of neural network but the recognition accuracy is less. For achieving better accuracy fusion of classifiers is recommended. In [25] author developed SVM classifier for character segmentation and k-NN classifier for character recognition. In this only on using benchmarking dataset better performance can be achieved. So, the character segmentation and character recognition plays a significant role in converting paper format script into digital format. The outline about the Devanagari script was described below.

III. BACKGROUND INFORMATION OF DEVANAGARI

Originated from the Brahmi script and the mother script of many Indian languages, Devanagari is used as reading and writing. The Devanagari is extensively spread over a wide belt of India. The gradual progression of the Devanagari from Brahmi is specified as below,

Brahmi script → bharati script (Bhagwat (Gita)) → Gupta Script → Nagari script →Devanagari script

To write many language of India, Devanagari is utilized in recent years such as Nepali, Konkani, Prakrit, Sindhi, Rajasthani, Marathi, Hindi and Sanskrit. However, it is also utilized as the secondary script for Punjabi and Kashmiri languages. The Indo-Aryan family of languages are considered as the Hindi, Sanskrit and Marathi. Additionally, the Sanskrit is defined as the mother of Hindi and traditional languages. In the India, Hindi is mostly spoken languages and it also considered as the official language of India. In the southern regions of India, Marathi is mostly spoken such as nager, Dadra and Goa etc. Generally, Devanagari consists of distinctive features of characters and word formation, methods to process the linguistic structures and well-constructed grammar rules. The concepts of the Devanagari is presented below,

The alphabet of the modern Devanagari script consists of 13vowels, 33 consonants along with 5 extra consonants and 10 numerals such as 0 to 9 whereas the alphabet of the modern Bengali. Devanagari consists a very large set of consonant clusters and ligatures, the ligatures are created by combining two or more characters from the Devanagari. Devanagari compound characters are ‘क्ष’, ‘ज्ञ’, ‘त्र’, ‘क्त’, ‘त्त’, ‘द्व’, ‘श्र’, ‘द्ध’ and a special three letter combination symbol is ‘ॐ’ (‘ओ३म’). In the Devanagari script, the word is developed with group of letters, which contains of different combinations of consonants and vowels in sequence. In the utilization of the compound characters to form a word, the complexity of word formation is increased. The Devanagari script has different unique properties and contains different measures in hits writing and reading styles which are described below,

- It is written top to bottom, left to right and read in the same way of order of sentence
- It does not have small characters or capital characters
- Additionally, it does not have spelling arrangement of letters
- It develops syllabic alphabets through combining characters and syllables.
- Different Devanagari symbols follow a phonetic order
- The halant is defined as the angled sub stroke character which a one of the consonant without any vowel.
- The Devanagari script has the unique features that are use a long continuous horizontal top line on the characters. The top line is named as the shirorekha. The characters are joined continuously to formation of word after that top line of all the characters are combined together one by one.
- In the Devanagari script, shirorekha is partially absents in different consonants and their words such as ‘भ’, ‘थ’ and‘ध’.
- It has the geometric structures of many vowels and consonants which contain closed forms and small loops. At the condition of two vowels with closed forms which consists of four constants categories such as loop with small vertical bar, closed letter, loop with vertical bar and loop without vertical bar.
- The modifiers and graphical elements are used above or below, left and right the vowels/consonants or both to formation of word frame in Devanagari script.
- The independent vowel of Devanagari script does not use consonants on its right and left sides and it is used to show lonely at the initial of a word or after another vowel.
- The left or right side of the consonant, short/ long vowels are presented and conjunct modifiers applied on the upper and right upper sides.

IV. PROPOSED ARCHITECTURE FOR TEXT GENERATION SYSTEM

The proposed Devanagari text generation system using AMA is designed to accept scanned document images which pre-processes, segments, feature extraction and lastly recognition and text generation. Generally, the AMA architecture is consists of two phases such as training and testing phases. During the training phase, a known set of text document images are taken which are further processed to get Devanagari characters.

When top lines are cleared from words during segmentation process, then independent and isolated Devanagari characters are obtained. In the segmentation process, the Devanagari characters can either be non-modified simple characters or those characters whose modifiers are separated. After that, the segmented character features are extracted for training the AMA algorithm. Initially, the AMA is calculated the minimum distance among the character and its upper/right/left modifier. After that, the recognition and text generation is attained with the help of AMA algorithm. During testing phase, the performance of the AMA algorithm is tested with an unknown set of document images. The proposed method architecture is illustrated in the figure 1.

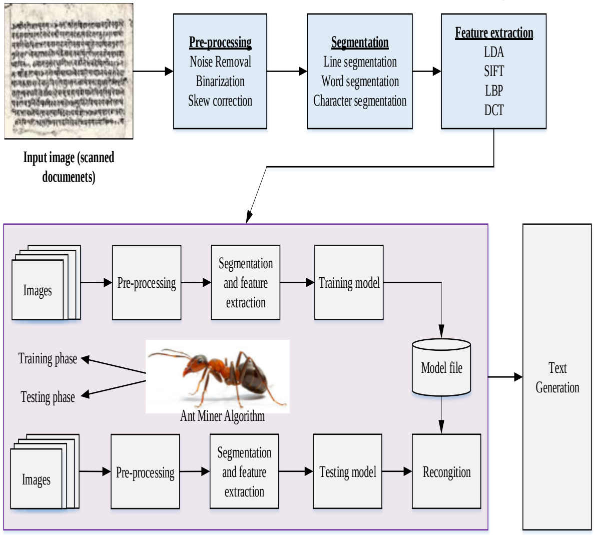


Figure 1: Architecture of the proposed method

Process of optimally recognizing a Devanagari script and text generation usually begins with an image document which is generally collected by scanning or via camera captures. To recognize and text generation of Devanagari script, document image are collected by camera, the images are required to pre-processes before they can be utilized for process of recognizing the character in that image and text generation. The pre-processing steps included noise removal, skew detection and binarization. At initial stage is binarization which image is converted into a grey scaled image from a coloured image. If image is a binary format, the pixel values of the image are either a 1 or a 0. 1 is considered as the black and 0 is considered as white. To correct skewness of the image, binarization is used in the architecture. The input document images are collected from the scanner, so the image tends to have a degree of skewness and it need to be corrected so as to have a less amount of error. The technique is used for correction of skewness in the image and which depends on the type of language. Compared to the different scripts, Devanagari get an easy because the words have shirorekha/headline, it is connects all the components. After done the two steps, the document Devanagari script image is processed by segmentation process to analyse the structure of the document. The segmentation process is used to finding out all the components in the document. With the utilization of the segmentation process, the document may be contains the different blocks of text, tables and images together which are finding from the document. The segmented characters and words are sending to the feature extraction segments. The extracted features are forwarded to the AMA algorithm recognition and classified. The recognized characters are automatically generated with different styles. The steps of the proposed method is presented below,

4.1. Pre-processing:

In the pre-processing stage, the input scanned document image is taken for the process. The main operation of the pre-processing is to systematize the word and remove variations among the input image. The pre-processing phase of this work consists of Noise removal, binarization and skew correction. The noise reduction can be used for remove noise from the image b filtering. The binarization process is used to binarize the image by thresholding method. The binarization process, mean value is considered and the above mean value taken as white and below mean value taken as black. Generally, the matric form black is 1 and white is 0.Additionally, skew is corrected and image is normalized. After that, the image is converted from RGB (red, grey, black) to gray level image through which binary image is obtained.

4.2. Segmentation:

The output image of the pre-processing method, the image is segmented further to extract the lines, words and characters by using the project profile. With the utilization of the horizontal projection profile, the lines are detected and located in bounding boxes. This is performed due to text lines consist high density of black pixels as compared to the gaps existing among the adjacent lines. After such line detection, similarly words are detected and located in boundary boxes with the utilization of the vertical projection profile. With the concept of horizontal profiling, the top lines are removed from the image word by word; the characters and upper/left/right/ modifiers are located and detected. From this, the top line of the each word is located and all pixels values made to zeroes because it consists the highest density of black pixels in a word. Finally, the characters, other components and modifiers are bounded in the boxes. Based on this Devanagari script characters (line, word and characters) are obtained.

4.3. Feature extraction:

Feature extraction method becomes a key aspect in achieving high recognition performance since these features plays major role in Devanagari text recognition and generation. In the feature extraction process, Devanagari images are extracted. In the images, the values are represented in the (rows × columns) matrix here; number of columns is equal to number of rows. The features are extracted from the image, using the different feature extraction techniques such as SIFT, LBP, LDA and DCT features. With the utilization of the different feature extraction methods, the features are extracted from the segmented Devanagari images.

4.3.1. SIFT

The SIFT feature extraction is used to extracts invariant feature based on invariant descriptor which was developed by Lowe in 2004. Generally, the SIFT feature remains invariant to noises, brightness variation, scaling, rotation and image translation. SIFT feature description mainly consists two main steps such as compute direction parameter of feature points and utilize graphical information around feature points to develop 128 dimensional descriptor.

Computation of direction parameter

To develop rotated invariance of descriptor of feature points, it can be compute the main direction of feature points in addition generate SIFT feature descriptor at this main direction.

The detected feature points can be defined as, finite difference computation shall be applied to find out pixel gradient module (S) and gradient angle amplitude (θ) in the region with center feature point. The mathematical formulation of this case is presented follows,

$$S(a,b) = \sqrt{T_1^2 + T_2^2} \qquad (1)$$

$$T_1 = T(a+1,b) - T(a-1,b) \qquad (2)$$

$$T_2 = T(a,b+1) - T(x,y-1) \qquad (3)$$

$$\theta(a,b) = Arc\tan\left(\frac{T(a+1,b) - T(a-1,b)}{T(a,b+1) - T(x,y-1)}\right) \qquad (4)$$

Where, $T(a,b)$ can be described as the image grayscale of the feature point at (a,b) on its scale. After that, histogram to statistically state pixel gradient module and direction in this region. The ordinate axis is accumulated value of gradient module and abscissa axis of histogram is the angle of amplitude of gradient direction which related to gradient direction angle. The gradient direction of the image is divided into 36 columns with the range of 0^0-360^0 and each column have the 100. With the utilization of the SIFT, some of the features are extracted from the input Devnagari script.

4.3.2. LBP

The operation in LBP is based on the threshold value which is denoted by the centre pixel value and the value of the adjacent pixel within the 3×3 matrix. Based on the threshold value the pattern in the form of binary system will be created. This binary pattern will denotes the feature of the texture. The LPB can be given by the following equation.

$$LPB(a_i,b_i) = \\ \sum_{m=0}^{7} 2^m\, X\left(L_m - L(a_i,b_i)\right) \qquad (5)$$

In the above mentioned equation, $LPB(a_i,b_i)$ denotes the LPB value of the pixel in the centre. $L(a_i,b_i)$, $L_m$denotes the value of the centre and the adjacent pixel respectively. m Denotes the index of the neighbouring pixel. The function $X(a)$ will become 0 if $a < 0$ and the $X(a) = 1$ in case if the$a \geq 0$. The finding of LPB value is illustrated with an example. The value of centre pixel is taken as 54 and it also represents the value of the threshold. In case if the value of the neighboring pixel is less than the threshold value, then the value of the function will be equal to zero. The value of the function becomes 1 if the value of the threshold is greater or equal. By means of scalar multiplication among the weight metrics and binary the value of LPB is calculated. And this value of LPB will be obtained in the 3×3 matrix. By varying the centre pixel and the location of the neighboring pixel many authors have developed different types of LPB which includes center symmetric LPB, circular LPB, and advanced LPB.

4.3.3. LDA

The discriminative feature was extracted by means of this LDA and it can be used for classifying the pattern. By means of label data the LDA understands the discriminative projection. This helps in reducing the distance within the class and increases the distance between classes in order to enhance the precision of the pattern classification. The projection vector is found by means of this LDA which reduces the distances between the similar class sample and increase the distance between the various class samples. The projection vector is calculated using the fisher criterion and the equation is represented below.

$$a = \arg\max \frac{a^T s_b a}{a^T s_w a} \qquad (6)$$

The eigenvalues based features are extracted in the Devanagari script. Single projection vector will be not sufficient to calculate the multiple classes and a group of projection vector is necessary to perform the multiple classes. After that, DCT is used to extract the features from the Devanagari script in Marathi.

4.3.4. DCT

DCT is a feature extraction technique and it will occur in two levels. In the first level, upon the entire image the DCT is given in order to gather the DCT coefficient. In the second level from the gathered DCT coefficient little of the coefficient were chosen to create the vector for the feature. The dimension of data is not reduced by DCT. It will only compact the data in the signal into a smaller coefficient. The DCT coefficient for the matrix $m \times n$ is calculated by the following equation.

$$F(a,b) = \frac{1}{\sqrt{mn}}\delta(a)\delta(b)\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} f(u,v)\ \mathrm{X}\cos\frac{(2u+1)a\pi}{2m}\ \mathrm{X}\cos\frac{(2v+1)b\pi}{2n} \qquad (7)$$

Where $a = 0,1,\ldots\ldots m\ , b = 0,1,\ldots\ldots n$
$\delta(\gamma)$ is given by

$$\delta(\gamma) = \begin{cases} \frac{1}{\sqrt{2}} & \gamma = 0 \\ 1 & otherwise \end{cases} \qquad (8)$$

In the above mentioned equation $f(u,v)$ denotes the intensity function of the image and the $F(a,b)$ denoted the DCT coefficient in matrix format. By means of this entire image the DCT coefficient frequency will be obtained. Basically the coefficient of DCT will be based on three bands which include the middle frequencies, lower and upper frequencies. Middle frequencies coefficients contain useful information and construct the basic structure of the image, high frequencies represent noise and small variations and the Low frequencies are linked with the illumination conditions. The different features are extracted with the help of SIFT, LDA, DCT and LBP. The extracted features are send to the input of the AMA algorithm.

4.4. Recognition and text generation

The extracted features are sending to AMA algorithm for training process. The AMA algorithm is mainly used for Devanagari character recognition and text generation from the document images. The AMA algorithm is working based on the training phase and testing phase. In the training phase, the images are send to processes of pre-processing, segmentation and feature extraction, after the images are send to training phase of the AMA algorithm.

After that, testing phases, the images are feed to input; in this phase Devanagari character is recognized. Finally, the text generation of Devanagari character is achieved with the help of the proposed method. The proposed AMA algorithm is used to recognition and text generation of Devanagari character. The process of the AMA algorithm is presented in the below section.

V. ANT MINER ALGORITHM

In the proposed method, the AMA algorithm is used for recognition and text generation of the devanagiri script. The AMA algorithm is working based on the training and testing stages and it also used as a classifier. The input document of the devanagiri script is send to the AMA algorithm for recognition and text generation. The general behaviour of the AMA algorithm is described in this section. This AMA was designed based on the natural behaviour of real ants. The ants lack of visual communication. But they always find a shortest path between the nest and the food source. This is achieved through a chemical substance called pheromone. This helps the ant to communicate with each other. A pheromone trial is created since as the ant moves a definite amount of pheromone is left on the ground. This makes more ants to follow the given trial. The probability that the ant selects a path is directly proportional to the number of ant already passes by the similar path. The main aim of the ant miner algorithm was to extract from the data the classification rules [26, 27].

The discovery of classification rules from the ant colony optimization is called as ant miner. This algorithm almost covers all the training cases. A training set is initialized in order to initialize all training cases in the ant miner. While initializing the rule list will be an empty list. Further an outer loop is executed where a classification rule is developed on all iteration [28].In inner loop the first step is to start all trials with similar quantity of pheromone. This indicates that all paths have similar probability to be selected by the ant. Next is the inner loop and it has three steps which includes as follows,

- ❖ Comprising rule construction
- ❖ Rule pruning
- ❖ Pheromone updating

At first the ant begins with a rule which has no terms in its precursor and then add one term at a time to its present fractional rule. The present fractional rule constructed by an ant denotes the current path followed by the ants. In similar way the term option added along the current partial rule denotes the direction in which then path will be prolonged. The term option needed to be added to the current partial rule depends on both of the quantity of pheromone related with each term and on the problem-dependent heuristic function [29]. To the current partial rule ant keeps joining single rule at a time until the stopping criteria is met. The criteria are as follows,

1. A min case per rule is defined as the lesser amount of cases covered than the user-specific threshold when any of the term is added to the rule. Previously all of the attributes are used by the ants. So there is no more attributes left to be joined to the rule. Each of the attributes must be used only once in the rule.

2. In order to remove or extract irrelevant terms from the rule constructed by the ant pruning of the rule is done. The irrelevant terms are included in the rule because of the using of local short-sighted heuristic function or may be due to stochastic variation in the procedure of term selection. The heuristic function removes attributes interaction and comprises only one attribute at a time.

3. In each trial the pheromone is updated since the pheromone trial in the path followed by the ant increases. But the pheromone in the other trial decreases (this simulated the evaporation of pheromone). In order to guide its search new quantity of pheromone is used by the ant to construct its rule.

4. In case of completion of REPEAT-UNTIL loop, best rule among the rule constructed by every ant is joined with the discovered rule list. Further from the training set the training cases that are correctly covered by this rule. The above described process is done when the number of training cases is larger than the user-specified threshold which is called as the Max uncovered cases. By means of reinitializing all the trials with the same quantity of pheromone the system will starts the new iteration of WHILE loop [29,30].

Comprising rule construction

Ant begins with an empty rule and keeps joining one term at a time to its current partial path in the way of already existing path. The term ij denotes the rule conditions where $A_i = V_{ij},A_i$ is the ith attribute and the V_{ij} is the jth value of domain ofA_i. To the current partial rule the probability that the term ij will be selected is given by the following equation.

$$p_{ij} = \frac{\vartheta_{ij}.\tau_{ij}(n)}{\sum_{i=1}^{a} x_i \sum_{j=1}^{b} (\vartheta_{ij}.\tau_{ij})} \qquad (9)$$

Where, ϑ_{ij} denotes problem dependent heuristic function value for the term ij , $\tau_{ij}(n)$ denotes the quantity of pheromone lined with the termij at the iteration$t$, a denotes the total number of attributes,x_i denotes one of the attribute not used by the ants or zero. b Denotes the number of values in the domain of $i\ th$ attribute [30].

Rule pruning

Many irrelevant terms occur during the rule construction because of the heuristic function and stochastic variation. So in order to remove irrelevant terms pruning is done. The main aim of this rule is to extract the irrelevant terms that are undeservedly included in the rule. This rule pruning has the ability to remove the over-fitting to the training data and improves the predictive power of the rule. The basic idea behind this pruning is to remove one term from the rule and this may improve the quality of the rule. The repeating of this process is done till the rule has only one term or there is no term which helps to improve the quality of the rule [30].

Pheromone updating

In each trial the quantity of pheromone is updated by increasing the pheromone in the trail followed by the ant and by decreasing the amount of pheromone in other trials. The quantity of pheromone initially deposited at each position is inversely proportional to the values of the attributes and it is given by the following equation [31],

$$\tau_{ij}(n=0)=\frac{1}{\sum_{i=1}^{a} b_i} \qquad (10)$$

In the above equation a denotes the total number of attributes, b_i denotes the possible number of values that can be taken by the attributes. The ant constructs its rule and that rule is being pruned, the quantity of pheromone in all paths must be updated. The updating follows two vital ideas which are as follows,

❖The quantity of pheromone related with the term ij that occurs in the rule is found by the ant is increased in proportion with the quality of that rule.

❖The quantity of pheromone related to the term ijthat does not occur in the rule is decreased by the simulation of evaporation of the pheromone.

❖The number of rule constructed is greater than or equal to the user specified threshold. The current ant constructs a rule that is similar with the rule constructed by the previous ant. The quality of rule is given by the following equation [31, 32],

❖ $$Rule\ quality = sensitivity * specificity \qquad (11)$$

❖ $$Sensitivity=\frac{TP}{(TP+FN)} \qquad (12)$$

❖ $$Specificity=\frac{TN}{(TN+FP)} \qquad (13)$$

❖ $$Rule\ quality = sensitivity * precision \qquad (14)$$

❖ $$Sensitivity=\frac{TP}{(TP+FN)} \qquad (15)$$

❖ $$Precision=\frac{TP}{(TP+FP)} \qquad (16)$$

❖In the above equations,

❖TP (True positive) is the number of cases that is equivalent to the rule antecedent (attribute value) and also equivalent to the rule consequent (class) this is the correct prediction.

❖TN (True negative) is the number of cases that is equivalent to the rule antecedent but not equivalent to the rule consequent this is an incorrect prediction.

❖FN (False negative) is the number of cases that is equivalent to the rule antecedent and not equivalent to the rule consequent this is an uncovered cases caused by a specific rule.

❖FP (False positive) is the number of cases that is not equivalent to either rule antecedent or rule consequent this is caused by rule antecedent and is specific to consequent class [32].

The process of the AMA algorithm and behaviour is presented in this section. With the utilization of the AMA algorithm, the recognition and text generation is achieved for Devanagari script. The structure of the AMA with proposed method is illustrated in the figure 2.

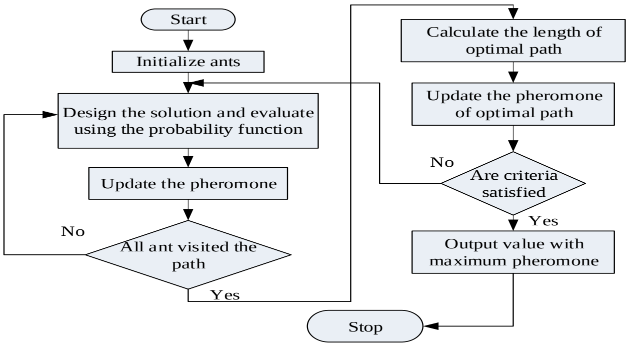

Start
Initialize ants
Design the solution and evaluate using the probability function
Update the pheromone
No
All ant visited the path
Yes
Calculate the length of optimal path
Update the pheromone of optimal path
No
Are criteria satisfied
Yes
Output value with maximum pheromone
Stop

Figure 2: Overall structure of the proposed method

At initially, the Devanagari script document data sets are collected from the open source system. In the proposed AMA algorithm is used for recognition and text generation of the devanagiri script. The datasets are divided in to two parts such as training stage and testing stage. The training stage, the pre-processing, segmentation and feature extraction is attained based on the process of different methods which are presented in the above section. The extracted features are sending to the AMA algorithm for training it. Following that the training phase gets initialized where $(n=1, i=1)$. The number of training cases must be greater than the user specified threshold if not the loop gets ended. If so the loop gets started by initializing all trials with the same quantity of pheromone. The maximum number of iteration condition need to be checked for recognition. So, the number of iteration must be greater than or equal to the number of ants $(n \geq no.of\ ants)$ and number of path followed must be greater than or equal to the number of rules constructed $(i \geq no.of\ rules\ constructed)$. Once the condition gets satisfied, construction rules are developed in the AMA algorithm for recognition. In the AMA algorithm, to remove the irrelevant terms, pruning process can be take place. After pruning process, the updating process of the AMA algorithm is attained based on the pheromone updating process. After updating process of the AMA algorithm, the best rule is selected form the developed rule of AMA algorithm for recognition. Based on the testing phase, the best solution achieved for recognition phase. Finally, based on the recognition, the text generation of the Devanagari script is attained. With the utilization of the proposed method, the devanagiri script recognition and text generation is achieved. The performance of the proposed method is analyzed in the below section.

VI. RESULTS AND DISCUSSION

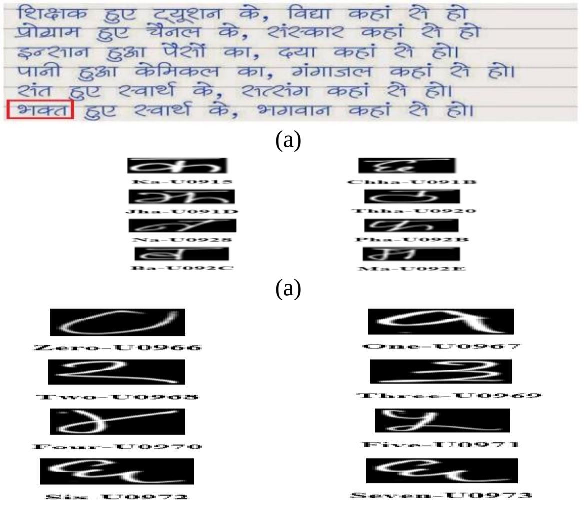
In this section, the performance of the proposed method is analysed. The experimental analysis of the proposed method has been carried out utilization the handwritten Devnagari script in Marathi. The experiment was completed on a dataset containing of 46 classes, each class consists of 1700 image samples. Based on the images, total images are considered as 78200 which used for analysis purpose. The Devanagari dataset was divided randomly 75 to 20 ratio for training and testing in AMA algorithm for recognition and text generation of Devanagari scripts.

To analysis the performance of the proposed method, it is compared with the existing methods of ANN and KNN. The proposed method is implemented in the Python platform and performance analysed with the utilization of the statistical measurements of accuracy, precision and recall. The statistical measurements are used for prove the efficiency and performance of the proposed method. The implementation parameters of AMA classifier is presented in the table 1.

Table 1: Implementation Parameters of proposed method

S.No	System	Description	Value
1	Dataset	Total images	78200
2		Classes	46
3		Each class images	1700
4	AMA algorithm	Number of ants	1000
5		minimum number of cases per rule	10
6		number of rules in test convergence	40
7		Exploitation rate	0.02

Initially, the dataset of Devanagari script is collected, after that pre-processing is started to removing noise, binarization and skew correction. The pre-processing images are send to the segmentation process. For effective recognition, from the images, line, word and character are separated. The separated images are used for feature extraction. The extracted features are mainly used for recognition, the features are feed to the AMA algorithm. The AMA can be classifier it work with two mode of process such as training and testing phase. From the features, 75 percentage of images are used for training phase and 25 percentage of images are used for testing phase. The training phases are consuming more percentage of image because how much effort to training the classifier it provided the best recognition results. After recognition of the Devanagari script from the handwritten document, the text can be generated with the help of the proposed approach. Finally, the performance of the proposed method is analysed with statistical measurements and it is compared with the existing methods of ANN and KNN respectively. The input handwritten dataset sample of the Devanagari script is illustrated in the figure 3.



शिक्षक हुए ट्यूशन के, विद्या कहां से हो
प्रोग्राम हुए चैनल के, संस्कार कहां से हो
इन्सान हुआ पैसों का, दया कहां से हो।
पानी हुआ केमिकल का, गंगाजल कहां से हो।
संत हुए स्वार्थ के, सत्संग कहां से हो।
भक्त हुए स्वार्थ के, भगवान कहां से हो।

(a)

Ka-U0915
Chha-U091B
Jha-U091D
Thha-U0920
Na-U0928
Pha-U092B
Ba-U092C
Ma-U092E

(a)

Zero-U0966
One-U0967
Two-U0968
Three-U0969
Four-U0970
Five-U0971
Six-U0972
Seven-U0973

(b)

Figure 3: Sample of dataset (a) handwritten document (b) characters with Unicode and (a) numbers with Unicode

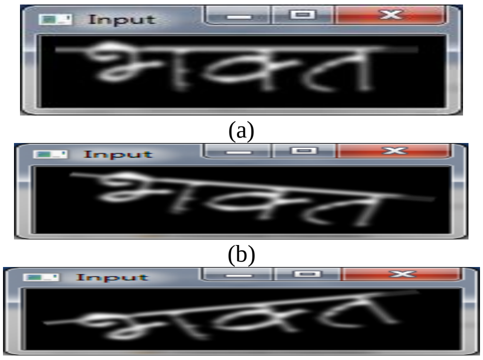
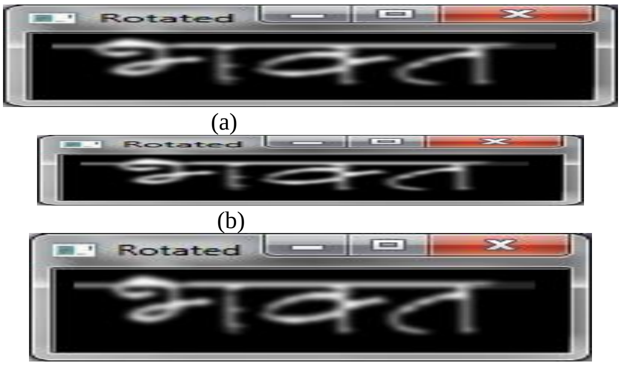


Figure 4: Input of the Devotee word (a) Straight image, (b) Left rotated image and (c) Right side rotated image

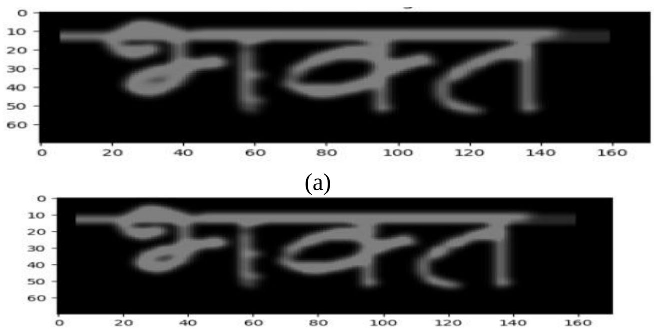
The handwritten document is consists of word and numbers so required to train the all characters and numbers in Devanagari Marathi script. The training of characters and numbers are used for recognition Devanagari script. Additionally, the handwritten input documents single word is taken for this analysis which means "Devotee" word in English it highlighted in the figure 3. The word input document is illustrated in the figure 4. The input image send to the pre-processing module, it contains three mode of operation such as noise removal, binarization and skew correction. The skew correction is used to correct the rotated image to the straight line image for recognition purpose. After that, the rotated image is used for binarization and noise removal purposes which reduces the noise from the image and adjust the thresholding value. The skew correction image of the Devotee is illustrated in the figure 5. From the figure 5, the skew correction output is analysed for different three input images. The skew correction is corrected the rotated image with straight line, the skew correction is an important concern for efficient recognition of Devanagari Marathi script word of Devotee.



(c)

Figure 5: Skew correction (a) straight image, (b) Left rotated image and (c) Right side rotated image

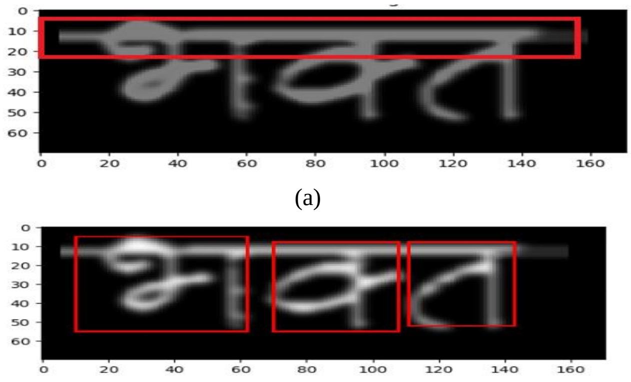
After that skew correction, the noise removal and binarization is takes place which also pre-processing techniques. The noise reduction and binarization images are illustrated in the figure 6. The pre-processing output images are send to the segmentation process which also segment the line, character and word. In the analysis, taking the word of devotee image, so omit the word segmentation. When, analysis the handwritten full document that also utilizes the line segmentation process. The segmentation outputs are presented in the figure 7.



(a)

(b)

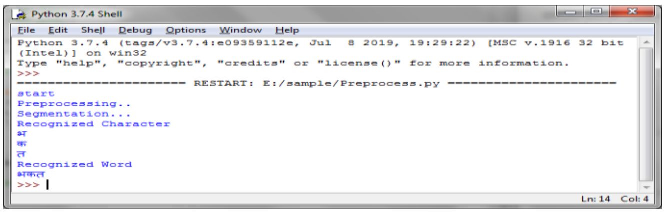
Figure 6: Analysis of (a) Noise Removal and (b) Binarization



(a)

(b)

Figure 7: Segmentation (a) Line Segmentation and (a) Character Segmentation



```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====================== RESTART: E:/sample/Preprocess.py ======================
start
Preprocessing..
Segmentation...
Recognized Character
भ
क
त
Recognized Word
भकत
>>> 
```

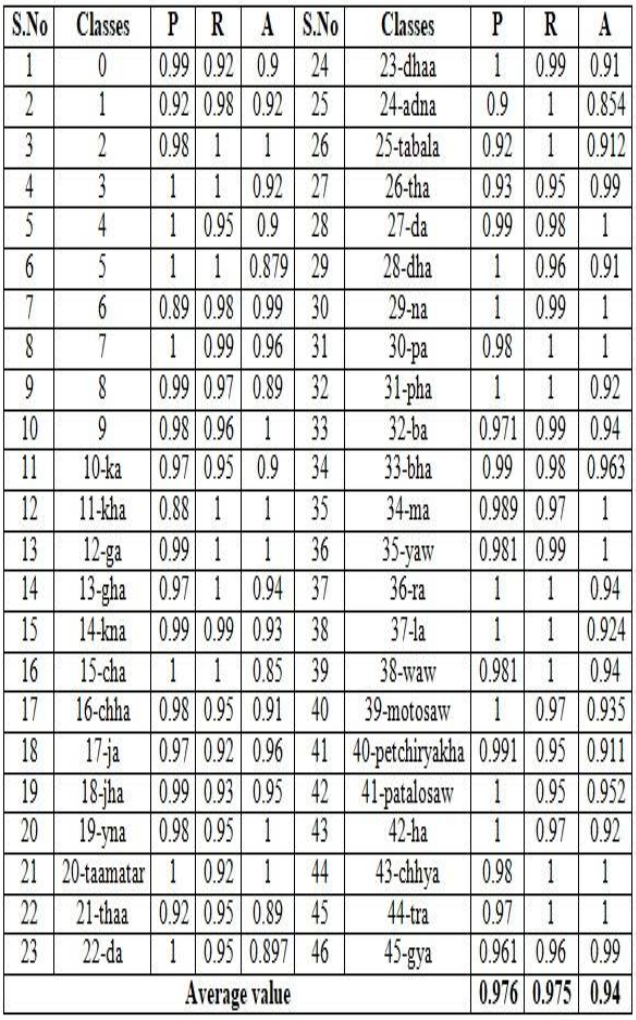
Figure 8: Final Recognition and text generation output

The segmented words are send to feature extraction techniques such as SIFT, LDA, LBP and DCT. The different features are extracted from the segmented image of Devanagari script. The features are used to training for the AMA classifier. Similarly, full document words are recognized with the proposed pre-processing, segmentation and feature extraction methods. Finally, proposed AMA classifier is used to recognition of Devanagari script. The final output recognition and text generation output is illustrated in the figure 8. With the utilization of the proposed method, the handwritten Devanagari script is recognized and text generated specially in Marathi script. The proposed method is compared with the existing methods of KNN and ANN respectively. The statistical measurements are used for analyse the performance of the proposed method and comparison analysis. The comparison analysis of the proposed method is presented in the below section.

6.1. Comparison analysis

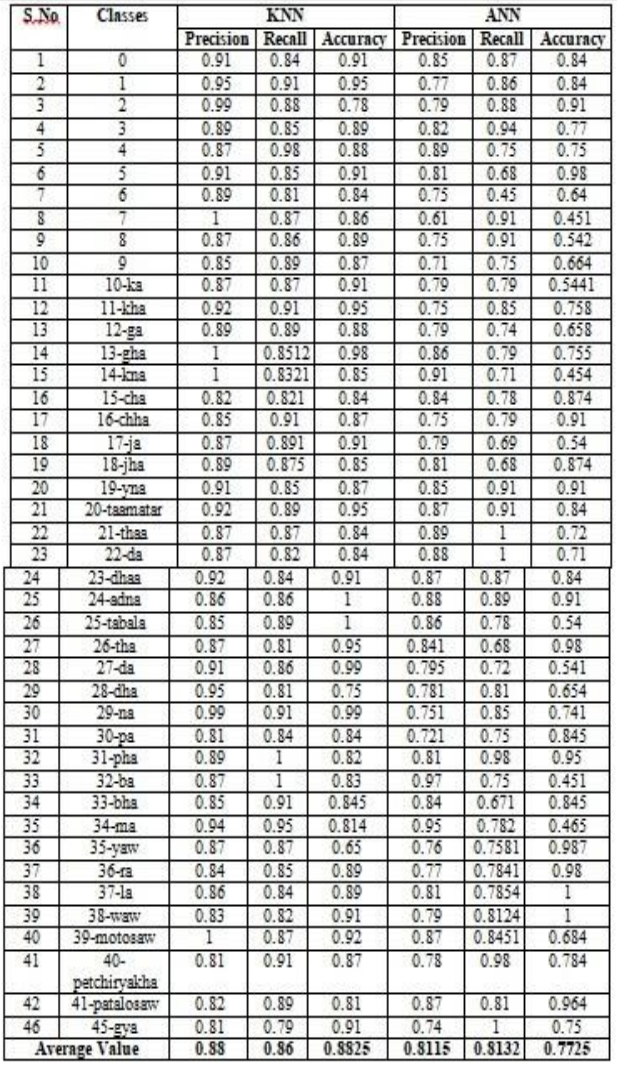
The selection of correct metrics for evaluating the performance of the system is vital to the result and the validation of the system. The parameters are selected by measuring the effectiveness of the processes involved in it. Precision, recall and accuracy are chosen to evaluate the performance of the proposed method. Based on the statistical measurements, can compute the performance of the proposed methodology. Implementation of the system is carried out using python. The proposed methodology is tested for Devanagari recognition and text generation in Devanagari handwritten document. 75% of the data in the dataset is used for training and the remaining 25% for validation. The statistical measurements are combining the performance matrix. In the proposed methodology three different performance matrixes are considered such as accuracy, precision and recall. The 46 classes' statistical measurements are analysed with accuracy, precision and recall performance matrices which presented in the table 2. Based on the statistical measurements analysis, the proposed method is evaluated and compared with the existing methods of KNN and ANN. The table 3 provides the descriptions of the existing methods measurements of recall, precision and accuracy. These values are compared with the proposed AMA algorithm.

Table 2: Statistical measurements of proposed method



S.No	Classes	P	R	A	S.No	Classes	P	R	A
1	0	0.99	0.92	0.9	24	23-dhaa	1	0.99	0.91
2	1	0.92	0.98	0.92	25	24-adna	0.9	1	0.854
3	2	0.98	1	1	26	25-tabala	0.92	1	0.912
4	3	1	1	0.92	27	26-tha	0.93	0.95	0.99
5	4	1	0.95	0.9	28	27-da	0.99	0.98	1
6	5	1	1	0.879	29	28-dha	1	0.96	0.91
7	6	0.89	0.98	0.99	30	29-na	1	0.99	1
8	7	1	0.99	0.96	31	30-pa	0.98	1	1
9	8	0.99	0.97	0.89	32	31-pha	1	1	0.92
10	9	0.98	0.96	1	33	32-ba	0.971	0.99	0.94
11	10-ka	0.97	0.95	0.9	34	33-bha	0.99	0.98	0.963
12	11-kha	0.88	1	1	35	34-ma	0.989	0.97	1
13	12-ga	0.99	1	1	36	35-yaw	0.981	0.99	1
14	13-gha	0.97	1	0.94	37	36-ra	1	1	0.94
15	14-kna	0.99	0.99	0.93	38	37-la	1	1	0.924
16	15-cha	1	1	0.85	39	38-waw	0.981	1	0.94
17	16-chha	0.98	0.95	0.91	40	39-motosaw	1	0.97	0.935
18	17-ja	0.97	0.92	0.96	41	40-petchiryakha	0.991	0.95	0.911
19	18-jha	0.99	0.93	0.95	42	41-patalosaw	1	0.95	0.952
20	19-yna	0.98	0.95	1	43	42-ha	1	0.97	0.92
21	20-taamatar	1	0.92	1	44	43-chhya	0.98	1	1
22	21-thaa	0.92	0.95	0.89	45	44-tra	0.97	1	1
23	22-da	1	0.95	0.897	46	45-gya	0.961	0.96	0.99
Average value							**0.976**	**0.975**	**0.94**

Table 3: Statistical measurements of existing method



S.No	Classes	KNN			ANN		
		Precision	Recall	Accuracy	Precision	Recall	Accuracy
1	0	0.91	0.84	0.91	0.85	0.87	0.84
2	1	0.95	0.91	0.95	0.77	0.86	0.84
3	2	0.99	0.88	0.78	0.79	0.88	0.91
4	3	0.89	0.85	0.89	0.82	0.94	0.77
5	4	0.87	0.98	0.88	0.89	0.75	0.75
6	5	0.91	0.85	0.91	0.81	0.68	0.98
7	6	0.89	0.81	0.84	0.75	0.45	0.64
8	7	1	0.87	0.86	0.61	0.91	0.451
9	8	0.87	0.86	0.89	0.75	0.91	0.542
10	9	0.85	0.89	0.87	0.71	0.75	0.664
11	10-ka	0.87	0.87	0.91	0.79	0.79	0.5441
12	11-kha	0.92	0.91	0.95	0.75	0.85	0.758
13	12-ga	0.89	0.89	0.88	0.79	0.74	0.658
14	13-gha	1	0.8512	0.98	0.86	0.79	0.755
15	14-kna	1	0.8321	0.85	0.91	0.71	0.454
16	15-cha	0.82	0.821	0.84	0.84	0.78	0.874
17	16-chha	0.85	0.91	0.87	0.75	0.79	0.91
18	17-ja	0.87	0.891	0.91	0.79	0.69	0.54
19	18-jha	0.89	0.875	0.85	0.81	0.68	0.874
20	19-yna	0.91	0.85	0.87	0.85	0.91	0.91
21	20-taamatar	0.92	0.89	0.95	0.87	0.91	0.84
22	21-thaa	0.87	0.87	0.84	0.89	1	0.72
23	22-da	0.87	0.82	0.84	0.88	1	0.71
24	23-dhaa	0.92	0.84	0.91	0.87	0.87	0.84
25	24-adna	0.86	0.86	1	0.88	0.89	0.91
26	25-tabala	0.85	0.89	1	0.86	0.78	0.54
27	26-tha	0.87	0.81	0.95	0.841	0.68	0.98
28	27-da	0.91	0.86	0.99	0.795	0.72	0.541
29	28-dha	0.95	0.81	0.75	0.781	0.81	0.654
30	29-na	0.99	0.91	0.99	0.751	0.85	0.741
31	30-pa	0.81	0.84	0.84	0.721	0.75	0.845
32	31-pha	0.89	1	0.82	0.81	0.98	0.95
33	32-ba	0.87	1	0.83	0.97	0.75	0.451
34	33-bha	0.85	0.91	0.845	0.84	0.671	0.845
35	34-ma	0.94	0.95	0.814	0.95	0.782	0.465
36	35-yaw	0.87	0.87	0.65	0.76	0.7581	0.987
37	36-ra	0.84	0.85	0.89	0.77	0.7841	0.98
38	37-la	0.86	0.84	0.89	0.81	0.7854	1
39	38-waw	0.83	0.82	0.91	0.79	0.8124	1
40	39-motosaw	1	0.87	0.92	0.87	0.8451	0.684
41	40-petchiryakha	0.81	0.91	0.87	0.78	0.98	0.784
42	41-patalosaw	0.82	0.89	0.81	0.87	0.81	0.964
46	45-gya	0.81	0.79	0.91	0.74	1	0.75
Average Value		**0.88**	**0.86**	**0.8825**	**0.8115**	**0.8132**	**0.7725**

From the table 2 and 3, the performance metric of accuracy, precision and recall can be analysed for proposed and existing models. Form the analysis of performance metrics, the proposed method have the values of 0.975, 0.974, 0.94 precision, recall and accuracy. Similarly, the existing model performance metrics are evaluated. From the evaluation, the precision of the proposed and existing methods of ANN and KNN are 0.975, 0.88 and 0.811. Compared to the existing methods, the proposed method is attain the best results for recognition and text generation of Devanagari script. From the evaluation, the recall of the proposed and existing methods are 0.974, 0.86 and 0.8132. Compared to the existing methods, the proposed method is attained the best results for recognition and text generation of Devanagari script. From the evaluation, the accuracy of the proposed and existing methods are 0.94, 0.8825 and 0.7725 Compared to the existing methods, the proposed method is attain the best results for recognition and text generation of Devanagari script. Finally, compared to the existing methods of ANN and KNN, the proposed method is attain the best recognition results. From the comparison analysis, the proposed method is proved the best recognition results.

VII. CONCLUSION

This paper proposed an AMA algorithm for recognition and text generation of Devanagari script in Marathi handwritten documents. The recognition and text generation of the Devanagari script can be achieved with the different process such as pre-processing, segmentation, feature extraction and recognition with text generation. The pre-processing consists of noise removal, binarization and skew correction; segmentation consists of word segmentation, character segmentation and line segmentation; Feature extraction consists of LBP, SIFT, LDA and DCT; Recognition and text generation is achieved with the help of the AMA algorithm. The proposed method is implemented in python platform and it is compared with the existing methods such as ANN and KNN. The proposed method is achieved 97%, 97% and 94% of precision, recall and accuracy in the recognition and text generation of Devanagari Marathi handwritten script. The ANN is achieved 88%, 86% and 88% of precision, recall and accuracy. Similarly, the KNN classifier is achieved 81%, 81% and 77% of precision, recall and accuracy. Therefore, the proposed method is attained best recognition and text generation of Devanagari scripts. This research work can be further extended for recognition of different Devanagari scripts for example Nepali with different feature extraction technique.

REFERENCES

1. Naveen malik and Aashdeep singh, “review of character recognition of offline handwritten Devanagari script”, an international journal of computer science and mobile computing, vol.5, pp-178-183, 2016.
2. Vijay Pal, Dhaka and Manoj Kumar Sharma, "An efficient segmentation technique for Devanagari offline handwritten scripts using the Feed forward Neural Network", an international journal of Neural Computing and Applications, vol. 26, no. 8, 2015.
3. Pratibha,Singh, , Ajay Verma, and Narendra S. Chaudhari. "Feature selection based classifier combination approach for handwritten Devanagari numeral recognition", vol. 40, no. 6 2015.
4. Pooja,Keshri, , Prabhat Kumar, and Rajib Ghosh, "RNN Based Online Handwritten Word Recognition in Devanagari Script.", In proceeding of International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 517-522, 2018.
5. Deepu,Kumar and Divya Gupta, "Review On Optical Character Recognition for Off-line Devanagari Handwritten Characters & Challenges", an international journal on computer science engineering and technology,vol.3 (2018).
6. M. S.,Sonawane, C. A. Dhawale, and Sanjay Misra, "Performance evaluation of classification techniques for Devanagari script," In proceeding of International Conference on Computing, Communication and Security (ICCCS), pp. 1-4., 2015.
7. Nitin Namdeo, Patil and Jayantrao Bhaurao Patil, "Performance Analysis of a Novel Text Watermarking Technique for Devanagari Text", an International journal on Intelligent Communication, Control and Devices, pp. 325-333, 2017.
8. Rajib,Ghosh and Partha Pratim Roy, "Comparison of zone-features for online Bengali and Devanagari word recognition using HMM", an International journal on Frontiers in Handwriting Recognition (ICFHR), pp. 435-440, 2016.
9. Soumen, Bag, and Gaurav Harit. "A survey on optical character recognition for Bangla and Devanagari scripts”, an international journal of economic source, no. 1 (2013).
10. Ved Prakash and Agnihotri, “Offline Handwritten Devanagari Script Recognition," an international journal on Information Technology and Computer Science, vol.8, no. 1 2012
11. Rajesh Kumar and Maurya, "Review of Advances in Digital Recognition of Indic Scripts”, an international journal of computer science, engineering and technology, vol.3, 2018.

12. Anilkumar N,Holambe, Ravinder C. Thool, and S. M. Jagade, "Printed and handwritten character & number recognition of Devanagari script using gradient features." an International Journal of Computer Applications , no. 9 2010.
13. Shailedra Kumar,Shrivastava and Sanjay S. Gharde, "Support vector machine for handwritten Devanagari numeral recognition" ,an International journal of computer applications , no. 11 2010.
14. S. D. Chitnis and Khanale, P. B, "Handwritten Devanagari character recognition using artificial neural network", an international Journal of Artificial Intelligence, vol.4, no. 1, pp.55-62 2011.
15. Munish,Kumar, R. K. Sharma, and M. K. Jindal, "Efficient feature extraction techniques for offline handwritten Gurumukhi character recognition", National Academy Science, vol. 37, no. 4 2014.
16. C. S. Yadav,Singh, Raghuraj, C. S. Yadav, Prabhat Verma, and Vibhash Yadav, "Optical character recognition (OCR) for printed Devanagari script using artificial neural network", an International Journal of Computer Science & Communication,vol.1, no. 1,pp.91-95,2010.
17. Rajiv,Kapoor, Deepak Bagai, and T. S. Kamal. "Skew angle detection of a cursive handwritten Devanagari script character image", an international Journal of the Indian Institute of Science,vol. 82, no. 3 & 4 2013.
18. Shruthi, Maher Sid-Ahmed,Kubatur and Majid Ahmadi, "A neural network approach to online Devanagari handwritten character recognition," an International Conference on High Performance Computing & Simulation (HPCS), pp. 209-214, 2012.
19. Satish and Kumar, "An analysis of irregularities in Devanagari script writing: A machine recognition perspective", an International Journal of Computer Science Eng, vol.2, no. 2 2010.
20. Santhosh kumar henge and B. Rama, " comparative analysis of OCR algorithms and invention analysis of character recognition approached methodologies", an international conference on power electronics, intelligent control and power systems, 2016.
21. Milind,Bhalerao, , Sanjiv Bonde, Abhijeet Nandedkar, and Sushma Pilawan,"Combined classifier approach for offline handwritten Devanagari character recognition using multiple features" ,In Computational Vision and Bio Inspired Computing, pp. 45-54, 2018.
22. Ghanahshyam B,Kshirsagar and Narendra D. Londhe. "Improving Performance of Devanagari Script Input-Based P300 Speller Using Deep Learning", IEEE Transactions on Biomedical Engineering (2018).
23. Ayan Kumar,Bhunia, Partha Pratim Roy, Akash Mohta, and Umapada Pal. "Cross-language framework for word recognition and spotting of Indic scripts," Pattern Recognition ,2018.
24. Pratibha,Singh, Ajay Verma, and Narendra S. Chaudhari,"Deep convolutional neural network classifier for handwritten Devanagari character recognition", In Information systems design and intelligent applications, pp. 551-561,2016.
25. Rul, Sahare, and Sanjay B. Dhok, "Multilingual character segmentation and recognition schemes for Indian document images", IEEE Access 6, 2018.
26. S. Marcus Karnan, Madhusudhanan and K. Rajivgandhi, "Fuzzy based ant miner algorithm in data mining for Hepatitis", an International journal on Signal Acquisition and Processing, pp. 229-232, IEEE, 2010.
27. Heitor S. Lopes, Parpinelli, Rafael S. and Alex Alves Freitas, "Data mining with an ant colony optimization algorithm" ,IEEE transactions on evolutionary computation 6, no. 4,2012
28. Ashraf M. Abdelbar and Salama, Khalid M, "Extensions to the Ant-Miner classification rule discovery algorithm", In proceeding International Conference on Swarm Intelligence, pp. 167-178, 2010.
29. Rafael S,Parpinelli, Heitor S. Lopes, and Alex Alves Freitas, "Data mining with an ant colony optimization algorithm", IEEE transactions on evolutionary computation 6, no. 4, 2012.
30. Thomas,Stützle, Manuel Lopez-Ibánez, Paola Pellegrini, Michael Maur, Marco Montes De Oca, Mauro Birattari, and Marco Dorigo, "Parameter adaptation in ant colony optimization," In Autonomous search, pp. 191-215,2011.
31. Marco, Dorigo and Christian Blum, "Ant colony optimization theory: A survey," Theoretical computer science, no. 2-3, 2015.
32. Fernando EB, Otero, Alex A. Freitas, and Colin G. Johnson, "A hierarchical multi-label classification ant colony algorithm for protein function prediction," Memetic Computing 2, no. 3, 2010.