International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-7 Issue-4s2, December 2019

# Character Recognition Using Hidden Markov Models

**Saish Bhende, Kutub Thakur, Jason Teseng, Md Liakat Ali, Nan Wang**

***Abstract:** The need for the character recognition in today's real time world has motivated us to do study on the process of HMM's character recognition. This technique is the finest and accurate one to get a word or a character or a line to be recognized almost at 100 percent success rate. This paper shows the different steps to be followed during the iterations for the recognition. The paper basically reflects the introduction to the procedure to the prediction to the typical errors to the success rate of the method.*

## I.INTRODUCTION

Characters can be a handwritten text by the human or a document generated by the machine. So character recognition has been defined as recognition of these type of text or document by the computer by certain methods. Character recognition can face difficulties because of different fonts available, different sizes available. In case of human written text it can be more difficult as each and every individual's handwritings vary. Little work has been done in text and character recognition from grey level images. The method we have used in this study is very simple and straightforward. We have chosen a method, where HMMs are used to describe each character class. The word is then matched with all possible combinations of models, using dynamic programming the string that gives the best match.

## II.GENERAL METHOD

Before going in details for the Hidden Markov Method, let us have a look on a very general method, which is followed while we go for the character recognition. The prime steps followed in the recognition of the words/characters are below: -

- Photo scanning of the text character-by-character
- Detection of lines of text
- Analyzed for light and dark areas in order to identify each alphabetic letter or numeric digit
- Detection of connected components
- Translation of the character image into character codes
- Compare the characters in the scanned image file to the characters in this learned set

**Manuscript published on 30 December 2019.**
***Correspondence Author(s)**
**Saish Bhende,** Research Scholar, Pace University, NY, USA.
**Kutub Thakur,** Assistant Professor, New Jersey City University, Jersey City, NJ 07305, USA.
**Jason Teseng,** Adjunct Professor, New Jersey City University, Jersey City, NJ 07305, USA.
**Md Liakat Ali,** Assistant Professor, Rider University, NJ , USA.
**Nan Wang,** Associate Professor, New Jersey City University, Jersey City, NJ 07305, USA.



## III.FACTORS AFFECTING RECOGNITION

There are certain factors that can affect the recognition of the characters and can cause the difficulties in the process. Few of the factors are listed below: -

- Wrinkled, torn, or otherwise damaged
- Faded or otherwise aged
- Discolored
- Smudged (or the text is otherwise obfuscated or distorted)
- Printed with low-contrast or colored ink
- Rendered with nonstandard fonts or in human handwriting, or printed on specific types of paper that decrease crispness and contrast between the background and foreground in the resulting scan
- Low pixel scanning

One of the major factors can be the scanning the document with inappropriate dpi. A document scanned under 200 dpi can cause poor results whereas document scanned above 600 dpi can cause more storage space.

## IV.LEARNED SET

The process of character recognition is fully dependent on the availability of the learned set. What is learned set exactly? So defining the learned set indicates the database we have for a particular character. Or in simple language, learned set is the container where there are plenty of size, style and fonts available for a particular character. More we have the data dictionary; more affective the method can run and more success rate we can achieve. To make a learned set, a text file needs to create with different fonts, size, and styles. Below is the example of a text file showing different characters.

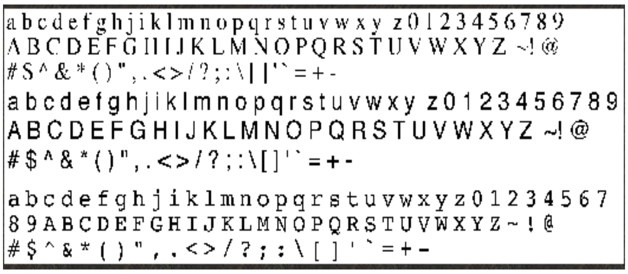


**Fig. 1: Learned Set**

## V.HIDDEN MARKOV MODELS

Hidden Markov models (HMM) together with element programming have been generally utilized as a part of the speech as well as character acknowledgment. Of late these strategies have additionally showed up inside the character recognition.

Two fundamental methodologies have been utilized, developing a Markov demonstrate either for every word or for every character. The last approach is decided for our application, since it doesn't require a constrained vocabulary.

A statistical tool used for modeling generative sequences characterized by a set of observable sequences.A HMM is a Markov chain for which the state is only partially observable.

A hidden Markov model is a doubly stochastic process with a basic stochastic process that is not detectable but rather must be seen through another arrangement of stochastic procedures that deliver grouping of perceptions. The shrouded Markov display requires three likelihood measures that are listed below: -

- The fundamental Markov chain A
- The perception probabilities B
- The initial probability vector

All the HMMs considered in this paper are left-to-right models with N states. Left-to-right models are pretty easy to train.

## VI.TRAINING

For each character class, the samples need to be labeled manually. The grey level images are then temporarily thresholder. Then the connected components are extracted labeled with the correct class labels.

A bounding box to determine the exact location of the image is used to bind the character. The features from the grey scale image are then extracted. A class description containing estimates of the parameters needed to define the hidden Markov model, is computed for each character, based on the extracted features.

## VII.FEATURE EXTRACTION

The location of the character from the word and its bounding box are used as a reference to the corresponding grey level image. After the feature extraction, the position and the size of the circumscribing box must be normalized.

The bounding box for all the characters should have the same height. During training the class membership of each character is known, and from this the position of the normalized box is found. The segmentation of the grey class image is done and then features are extracted for each sub image. For each pair of consecutive grey levels within a column, the minimum value is chosen. The simple features are extracted from that image. No complicated computations are required. Furthermore, the features are easily combined with the Markov models for the characters. We have not made attempts at making the features invariant to scale and rotation. A disadvantage of these features is that the amount of data will be relatively large and may increase the processing time needed for the classification. So, this is partly compensated with simple feature extraction.

## VIII.PARAMETER ESTIMATION

For each character model, the set of parameters c = (A, B, V) has to be estimated from a training set with a sufficient number of samples from each character class. In addition, the states per model must be defined. Ideally there should be equal number of models avoiding the manual design of the models. Experiences have shown that the geometrical probabilities are good whereas the exponential probabilities should be avoided as they are not considered to be good. So, it is good to define the model with the large number of states per model and are considered to be 6 states per model ideally. As none of the states should be skipped so there should be equal to or more number of frames than the number of states for the particular model. The number of states for the shortest characters are less. E.g.: - For “I” , “i” , “.” , “,” etc.

The transition probabilities are constrained to

$$a_{ij} = 0, j < 1, j > i + 1,$$

showing that none of the states should be skipped and the movement of the states are from left to right. Moreover, while going for the frames of the images, even the smaller part of the images should not be skipped. The initial probabilities of the states are: -

$$\pi_i = \begin{cases} 1\ if\ i = 1 \\ 0\ otherwise \end{cases}$$

The feature vectors are assumed to be continuous and are defined as: -

$$b_j(O_t) = \frac{exp^{-\sum_{d=1}^{D}(O_{td}-\mu_{jd})^2}}{(2\pi)^{D/2}}$$

Where D is the feature vector.

The equation uses k-means method for the calculation and estimations. The transition and prior probabilities were kept fixed during the training.

## IX.RECOGNITION

A sequence of feature vectors is extracted for an entire word at a time based on the grey level image. Viterbi Algorithm is used to match these sequences of frames against the HMM’s of the single character.

FEATURE EXTRACTION and LEVEL BUILDING involves the thresholding of the grey level image.We can contract or expand a HMM for smaller or larger part of the frames.The words are sorted according to their location in the document. For each model, it is checked if it is related to the current interval of frames. For each HMM c, level l we do a Viterbi match against the frames sequence and get the following details for each frame t

- ✓ P (l,t,c) where P is log probability
- ✓ B (l,t,c) where B is back pointer indicating the path

Maximization over c is performed which give best match for position t, the probability for this match and back pointer of the best character model.

## X.SYMBOLIC EXAMPLE

Symbolic examples are given by:

Example of word images generation for a two-character alphabet {‘n’, ‘u’}.

Set S = {Su1,..., Su5, Sn1,..., Sn5}

Set V = {V1, V2, V3, V4}



Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

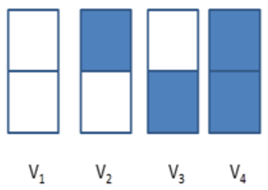


**Fig. 2: Vertical thin frames**

Assign to each Vi a thin vertical image:

The word model (for words like ‘unnunuuu’) is constructed by joining two left-right character models.

## XI.TRAINING

Samples from each character class are _rest manually labeled and gathering of training examples. Input is given by instances of character images. Output is given by a sequence of feature vectors per character instance.The grey level picture is incidentally thresholder, and associated segments are extricated and marked with the right class name. The characters are spoken to by their jumping box and area in the picture, and elements are extricated from the first grey level picture. A class portrayal containing assessments of the parameters required for the Hidden Markov model is registered for every character, in view of the removed elements.

### Feature Extraction

Transformation of the image into a sequence of 20-dimensional feature vectors. Thin overlapping rectangles split into 20 vertical cells. The feature vector for each rectangle is computed by computing the average luminosity of each cell. The area of the double character and its jumping box are utilized as a kind of perspective to the comparing dark level picture. Be that as it may, before the component extraction, the position and size of the surrounding box must be standardized. We require that this container ought to have a similar stature for all characters. Amid preparing, the class participation of every character is known, and from this the position of the standardized box is found. The grey levels of the sub picture, of the standardized box, are then aligned before the components are separated.

The components were basically decided for their effortlessness. They are effectively separated and strong, as no entangled calculations are required. Moreover, the elements are effectively joined with the Markov models for the characters. In our review, we have not made endeavors at making the components invariant to scale and pivot. Invariance to text styles can to a specific degree be acquired through cautious plan of the Markov models.

A burden of these components is that the measure of information will be generally extensive and may expand the handling time required for the order. Be that as it may, this will be somewhat made up for as the calculations required for the element extraction are exceptionally basic.

### Recogntion

The acquisition of the written mark by a sensory organ (sensor). The set of pre-treatments necessary to improve the quality of the acquired data. The process of segmenting the text into words or a finer segmentation of words into symbols Based on the grey level image, a sequence of feature vectors is extracted for an entire word at the time. A modeled Viterbi algorithm is used to match these sequences of frames against the HMMs of the single characters.

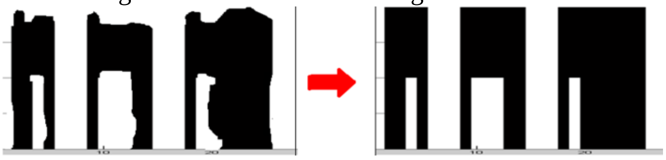

**Fig. 3: Recognition of characters in sequence**

### Clustering

The goal of a clustering analysis is to divide a given set of data or objects into a cluster represent subsets. Thepartition should have two properties.

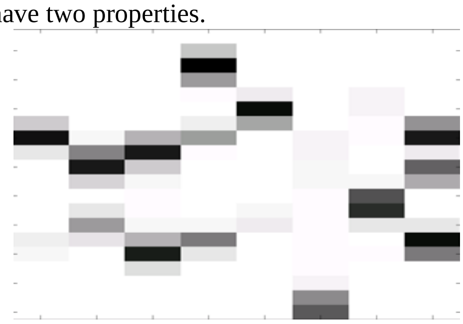

**Fig. 4: Clustering of the eight typical vectors**

A. Homogeneity inside clusters: The data, which belongs to one cluster, should be as similar as possible.
B. Heterogeneity between the clusters: The data, which belongs to different clusters, should be as different as possible.

Given a large set of feature vectors (100k) extracted from the training set of images and compute the k-means clustering with 512 clusters.

## XII.EXAMPLES OF GENERATED WORDS

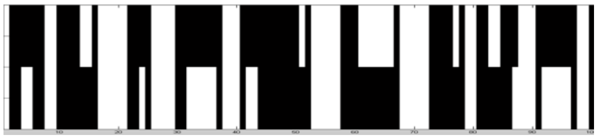

**Fig. 5: Example: word 'nunnnuuun'**

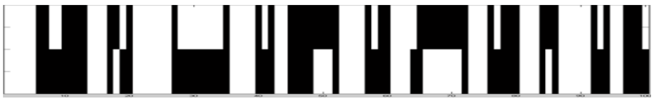

**Fig. 6: Example: word 'uuuununu'**

### Discretization

Discretization is considered as a divider that performs two essential operations the first task is to convert the value of the continuous characteristics into discrete. The second one is to divide the value and categorized them into appropriate intervals. The main objective of the discretization of the continuous characteristics is to represent the min a better way. There are some well-known techniques for discretization including Equal Information Gain, Maximum Entropy, and Equal Interval Width.



Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Another method proposed is the Invariants Discretization method, is proved to be better in efficiency by having higher accuracy and better rates of identification. The method is supervised type and starts by choosing the suitable intervals to represent the writer's information. The upper and lower boundaries are then set for each interval. The number of intervals for an image must be the same as the number of the feature vectors.


**Fig. 7: Feature extraction example**

If we have a set of basic patterns (thin images of the observation symbols), we can transform any sequence of thin images into a sequence of symbols the input for our HMM.We do not deal with images of thin slices directly but rather with some feature vectors computed from them (and then compare vectors instead of matching images). The basic patterns (feature vectors) can be found with k-mean clustering (from a large set of feature vectors).

### Feature Extraction

A grey level image is threshold, and from this binary image connected components are identified and grouped into words. The words are sorted per their location in the document. Based on the knowledge of the words' location, the further processing can now be performed on the original grey level image.

The bounding box of each word is normalized before the features are extracted, and the grey values of the resulting box are calibrated. At this point the class membership of the characters is unknown. Therefore, the normalization is performed relative to the baseline, which is found by investigating the grey levels of each scanline of the normalized box, starting from the bottom. When the average grey level goes below a limit and the difference from the average grey level of the scanline below is small, we assume that we have found the baseline. The limits are estimated from the grey levels of the image.

From the normalized box the features are extracted for each column, in the same way as during training, resulting in a list of frames. Before the frames are sent to recognition, empty frames are removed. By empty frames we mean the background frames between characters.

These are identified by investigating the minimum grey value of the frames. The reason for removing these frames, is that the training was performed only on single characters where there are no empty frames. Although the frames are removed, the information about where the empty frames were found is stored to be used later.

The location of the binary character and its bounding box are used as a reference to the corresponding grey level image.

However, prior to the feature extraction, the position and size of the circumscribing box must be normalized. We require that this box should have the same height for all characters. During training, the class membership of each character is known, and from this the position of the normalized box is found.

Features are extracted for each column of the sub image, and for each pair of consecutive grey levels within a column, the minimum value is chosen. The features were mainly chosen for their simplicity. They are easily extracted and robust, as no complicated computations are required.

Furthermore, the features are easily combined with the Markov models for the characters. Similar features, although extracted from binary images.

In our study, we have not made attempts at making the features invariant to scale and rotation. Invariance to fonts, can to a certain extent be obtained through careful design of the Markov models.

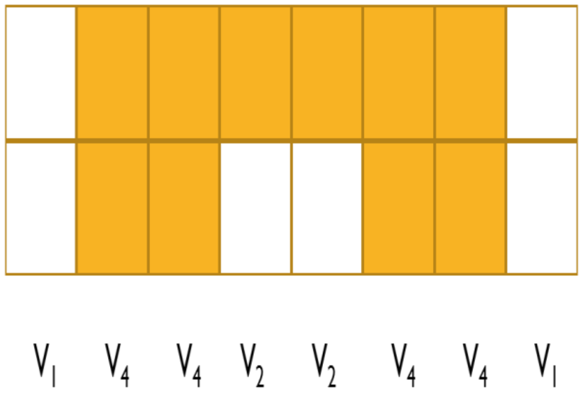


**Fig. 8: Word Model vertical image values**

A disadvantage of these features is that the amount of data will be relatively large and may increase the processing time needed for the classification. However, this will be partly compensated for as the computations needed for the feature extraction are very simple.

Transformation of the image into a sequence of 20-dimensional feature vectors. Thin overlapping rectangles split into 20 vertical cells. The feature vector for each rectangle is computed by computing the average luminosity of each cell.

## XIII.WORD MODEL

The word model generates a sequence of occurrence of characters' n and u there are two tables demonstrated as tables A and B.

**Table I: Probability values for n and u**

| A | Sn1 | Sn2 | Sn3 | Sn4 | Sn5 | Su1 | Su2 | Su3 | Su4 | Su5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Sn1 | 0.5 | 0.5 | | | | | | | | |
| Sn2 | | 0.5 | 0.5 | | | | | | | |
| Sn3 | | | 0.5 | 0.5 | | | | | | |
| Sn4 | | | | 0.5 | 0.5 | | | | | |
| Sn5 | **0.33** | | | | 0.33 | **0.33** | | | | |
| Su1 | | | | | | 0.5 | 0.5 | | | |
| Su2 | | | | | | | 0.5 | 0.5 | | |
| Su3 | | | | | | | | 0.5 | 0.5 | |
| Su4 | | | | | | | | | 0.5 | 0.5 |
| Su5 | **0.33** | | | | | **0.33** | | | | 0.33 |



*Published By:*
*Blue Eyes Intelligence Engineering*
*and Sciences Publication (BEIESP)*

International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-7 Issue-4s2, December 2019

Table II: Vertical image values for n and u

| B | V1 | V2 | V3 | V4 |
|---|---|---|---|---|
| Sn1 | 1 | | | |
| Sn2 | | | 0.05 | 0.95 |
| Sn3 | | 1 | | |
| Sn4 | | | 0.05 | 0.95 |
| Sn5 | 1 | | | |
| Su1 | 1 | | | |
| Su2 | | 0.05 | | 0.95 |
| Su3 | | | 1 | |
| Su4 | | 0.05 | | 0.95 |
| Su5 | 1 | | | |

## XIV.PREDICTION

Humans are said to unintentionally trace handwriting sequences in their brains based on handwriting experiences when recognizing written text. In this paper, we propose a model for predicting handwriting sequence for written text recognition based on handwriting experiences. The model is first trained using image sequences acquired while writing text. The image features of sequences are self-organized from the images using Self-Organizing Map. The feature sequences are used to train a neuro-dynamics learning model. For recognition, the text image is input into the model for predicting the handwriting sequence and recognition of the text. We conducted two experiments using ten Japanese characters. The results of the experiments show the effectivity of the model. We utilize a dynamical system for creating the prediction system. As we aim to create a model based on human’s recognition system, we focus on the following points. 1) Generalization from few handwriting experiences. 2) Adaptability to variations in writing patterns. Concerning the first point, we use neural networks for creating the model. Self-Organizing Map (SOM) is used for self-organizing image features from raw images. The use of a self-organizing model for extracting features (instead of predefined features) provides the capability to adapt to other applications. For the dynamical system, we use recurrent neural network. The generalization capability of neural networks provides the capability to adapt to unknown patterns from few training data. Concerning the second point, recurrent neural networks can create attractors which attract writing variations to stable patterns.

## XV.TYPICAL ERRORS

Errors are unavoidable in advanced computer vision applications such as optical character recognition, and the noise induced by these errors presents a serious challenge to down-stream processes that attempt to make use of such data. In this paper, we apply a new paradigm we have proposed for measuring the impact of recognition errors on the stages of a standard text analysis pipeline: sentence boundary detection, tokenization, and part-of-speech tagging.

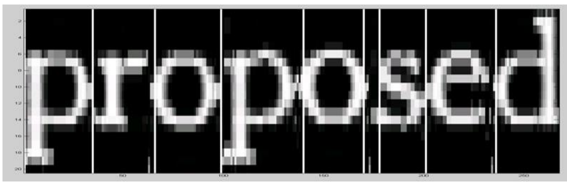

**Fig. 9: Predicted: proposled. We see that the there is a short strip between 'o' and 's' where the system predicted an 'l'.**

**Fig. 10: Predicted: dennocratic. The character 'm' was predicted as a sequence of two 'n' characters.**

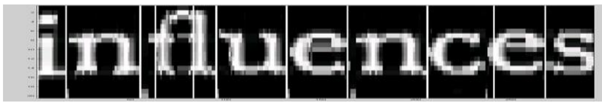

**Fig. 11: Predicted: inifluenoes. The system interpreted the character 'c' and the beginning of character 'e' as an 'o', and it predicted an extra ‘i'.**

Our methodology formulates error classification as an optimization problem solvable using a hierarchical dynamic programming approach. Errors and their cascading effects are isolated and analyzed as they travel through the pipeline. We present experimental results based on a large collection of scanned pages to study the varying impact depending on the nature of the error and the character(s) involved.

## XVI.RESULTS

HMM and Viterbi algorithm going hand in hand are considered to be very promising techniques for the character recognition weather it is a character or a word. The speed of this method is slow but that can be improved by reducing the dimensions and also the feature vectors.Moreover, the efficiency can be improved by incorporating the probabilities for the maximum characters. Studies have shown the accuracy rate of more than 98 percent while using the combination of both HMM and Viterbi.

## XVII.SUMMARY

The work exhibited in this paper demonstrates that the utilization of HMM with Viterbi coordinating and level building is a promising system for acknowledgment of content in dim level pictures. As of now the efficiency of the calculation is not succulent, and this will be a subject for further reviews. There are a few methods for expanding the speed of the acknowledgment, both by diminishing the measurement and number of highlight vectors and by making the execution of the calculation more efficient.

We feel that a superior outcome could have been acquired, if the Markov show for each of the characters had been physically planned. Additionally, consolidation of probabilities for trigrams of characters, may expand the acknowledgment rate.



Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

In this review, we have tried the technique on an information set brought just from one single archive. Later on, we might want to accomplish all the more testing on archives of changing dark levels and differentiation. We might likewise want to look more into the issues of invariance to scale and textual style.

## REFERENCES


1. J. Rocha & T. Pavlidis: A Shape Analysis Model with Applications to a Character Recognition System" IEEE Trans. Pattern Machine Intell., Vol. 16, No. 4, 1994
2. L. Wang & T. Pavlidis: Direct Gray-Scale Extraction of Features for Character Recognition" IEEE Trans. Pattern Machine Intell., Vol. 15, No. 10, 1993
3. F. R. Chen, L. D. Wilcox & D. S. Bloomberg: Detecting and Locating Partially Specified Keywords in Scanned Images using Hidden Markov Models" Proc. Sec. Int. Conf. Doc. Anal. Recog., pp. 133{138, 1993.
4. M-Y. Chen, A. Kundu& J. Zhou: OnLine Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network" IEEE Trans. Pattern Machine Intell., Vol 16, No. 5, pp 481-496, 1994.

C. B. Bose & S. Kuo: Connected and Degraded Text Recognition using Hidden Markov Model." Pattern Recognition, Vol 27, No. 10, pp. 1345-1363, 1994.

5. O. E. Agazzi& S-S. Kuo: Hidden Markov Model based Optical Character Recognition in the Presence of Deterministic Transformations" Pattern Recognition, Vol 26, No. 12, pp. 1813-1826, 1993.
6. L. R. Rabiner and S. E. Levinson: A Speaker-Independent, Syntax-directed, Con-nected Word Recognition System Based on Hidden Markov Models and Level Building." IEEE Transaction on Acoustics, Speech and Signal Processing, vol ASSP-33, No 3, p. 561-573.
7. L. R. Rabiner: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." Proceeding of the IEEE, vol 77, No 2.
8. G. Konheim: Cryptography: A Primer.", John Wiley, New York, 1982.




*Published By:*
*Blue Eyes Intelligence Engineering*
*and Sciences Publication (BEIESP)*