Mixed Language Based Offline Handwritten Character Recognition Using First Stroke Based Training Sets

Magesh Kasthuri magesh.kasthuri@wipro.com
Research Scholar
SCSVMV University, Kanchipuram, India

V.Shanthi drvshanthi@yahoo.co.in
Professor, Dept. of Computer Science
St. Joseph's College of Engineering
Chennai, India

Venkatasubramanian Sivaprasatham mukundmeghna@gmail.com
Professor, Dept. of Information Technology,
Nizwa College of Technology,
Nizwa, Sultanate of Oman.

Abstract

Artificial Neural Network is an artificial representation of the human brain that tries to simulate its learning process. To train a network and measure how well it performs, an objective function must be defined. A commonly used performance criterion function is the sum of squares error function.

Full end-to-end text recognition in natural images is a challenging problem that has recently received much attention in computer vision and machine learning. Traditional systems in this area have relied on elaborate models that incorporate carefully hand-engineered features or large amounts of prior knowledge.

Language identification and interpretation of handwritten characters is one of the challenges faced in various industries. For example, it is always a big challenge in data interpretation from cheques in banks, language identification and translated messages from ancient script in the form of manuscripts, palm scripts and stone carvings to name a few.

Handwritten character recognition using Soft computing methods like Neural networks is always a big area of research for long time and there are multiple theories and algorithms developed in the area of neural networks for handwritten character recognition

Keywords: Handwritten Character Recognition, Noise Reduction, Pre-processing Techniques In Character Recognition, Pattern Matching, Strokes, Fixed-language, Training Neural Networks, Gabor Filter.

1. INTRODUCTION

The key idea of this paper is broadly categorized as:

- Study and evaluation of various noise reduction techniques in Character recognition and establishing mechanism for properly identifying the base of noise reduction (eg: strokes, shapes, weightage, fonts etc.,) to handle mixed language character recognition.

- Defining an improvised training process called self-training based on first stroke identification.

- Design an algorithm for first stroke identification and further methods of segment identification in an offline character recognition.

- Conceptualize a unified system (system and methods) utilizing above training and character recognition process as a unique and combined Character recognition and interpretation system handling mixed language content.

2. PROBLEM DESCRIPTION

A stroke is not limited to a continuous line segment. A stroke may also include a portion of a character that has a discontinuity in its representation. For example, an English alphabet ‘i’ may also be considered as a single stroke according to some embodiments in spite of a discontinuity in its representation because there is no sudden change in angle in any portion of this alphabet. Therefore, there is need for greater accuracy in offline handwriting recognition of such handwritten text. Hence displaying the confidence of recognition helps the user to decide if this can be taken as acceptable threshold or improvise with further noise reduction or manual correction process.

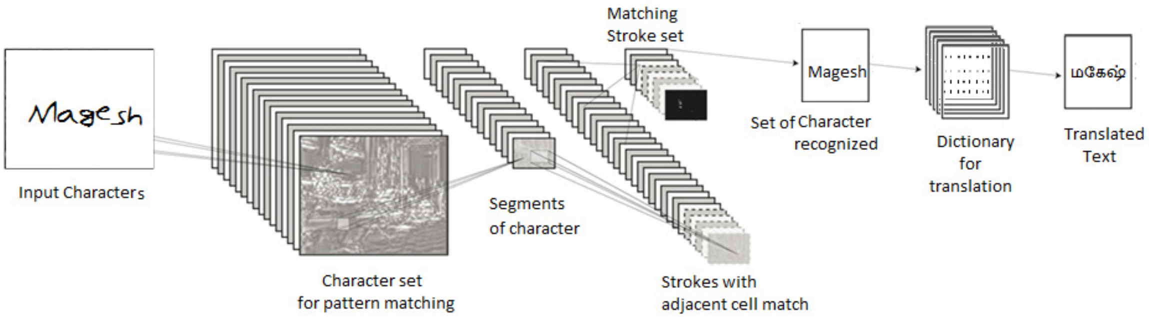


FIGURE 1: Proposed System for Handwritten Character Recognition.

One challenge associated with offline handwriting recognition is that the handwritten text cannot be edited or re-entered by the user until the entire handwritten text is recognized. A user, thus, may not be able to provide a feedback for correction after the recognition of each character or word in case of an incorrect recognition of that character or word. Thus, the errors in the offline handwriting recognition of that handwritten text may keep on accumulating in the absence of user supervision. Therefore, it is desirable to increase the accuracy of recognition of handwritten text in systems that implement offline handwriting recognition.

3. REVIEW OF EXISTING SOLUTION

The article in [1] discusses an Online character recognition designed based on Feature extraction from sample character using Gabor filter and Noise removal (size &shape normalization, imaginary strokes) is done prior to that. Pattern generation based on training with input feature vectors and Handwritten character Feature identification is done using Gabor filter.

Pattern generation and classification (statistical analysis) and Comparison of generated pattern with reference patterns to recognize the character are additional feature of the system proposed. Pattern based recognition is partially common with proposed approach whereas stroke identification is a novel part in pattern or relative cell forming in proposed system.

The article in [2] proposes a system designed for online character recognition (handwriting) suitable for noisy environments having - Improved accuracy based on self-learning specific to separate characters and users which is Alphabet independent and Low resource usage (40 KB) - mobile devices. It also uses Activity matrix based feature extraction & character comparison. The claim uses feature extraction like proposed system but the concept behind them is classical way of training the data which is completely different from our system.

Noise reduction (due to handwritten character stroke change) based on adjacent cell comparison from matrix of strokes and Binarization of strokes post feature extraction.

As per [10], Handwritten character recognition based on decomposition of characters into segments or features is designed which has Weightage based character recognition from features and Binarization of data based on feature comparison with highest score or Weightage from comparison. This comparison is between features and character model. The functionality includes decomposition the handwritten input character into one or more segments in accordance with the model specific segmentation scheme of the respective character model.

As per [11], Character Recognition for Ink based characters and uses mathematical notation for representing character shapes with Vector based recognition. Noise Reduction or feature filtering using mathematical comparison from vector representation and Repetitive process for vector normalization. The claim uses feature extraction like proposed system but the concept behind them is mathematical representation of shapes in the character system (predefined vocabulary through training or manual feed).

Method of language identification and language identifying module using short word lists and n-grams [21] does Pre-processing using n-gram technique (statistical representation) and Character Recognition based on knowledge base (training input). The Knowledge base is mapping to input source after feature extraction. This is a language identifying module and a method for identifying the language of a text string This is for providing language information to another application (eg: text-to-speech system in this case). This system uses language detection to feed the content to "text-to-speech” system whereas we use the detection for translating the content to target language.

4. PROCESSING COMPARISON WITH OTHER EXISTING SOLUTION

Performance of single-algorithm systems drops precipitously as the quality of input decreases. [3] In such situations, a human subject can continue to perform accurate recognition, showing only a gradual decrease in reliability. Collaboration between separate algorithms proves beneficial, in that such systems will allow a gradation of recognition levels expressed as probabilities or loose guesses to be passed from one level to the next. More specifically, a front-end system will perform some useful first-order basic processing. Then a second level of processing will be engaged which will judge whether to assimilate the results of the first process, extend them and proceed to the next stage with a positive recognition, or to dismiss them and reinvoke the first level again while asking for modifications.

In one embodiment[1] called Gabor filter based handwritten character recognition system, a character recognition method executed on an electronic device is disclosed, the method comprising: receiving, at the electronic device, an image representing a character including one or more central strokes; determining a set of parameters associated with each of the one or more relative (associative) strokes; comparing, for each of the one or more relative strokes, the associated set of parameters with a plurality of stored sets of adjacent parameters, wherein each of the plurality of stored adjacent strokes is associated with a stored set of relative parameters; identifying next stroke, from among the plurality of stored strokes, corresponding to each of the one or more strokes based on the comparison to identify the possible character comprising these strokes in order

5. STEPS INVOLVED IN TRAINING

The multiple-layered system which makes up any robust handwriting recognizer has progressed greatly from the days when character recognition meant reading printed numerals of a fixed-size OCR-A font. However, only in a decade have the successes within the field approached the level of a truly practical handwriting recognizer [1].

If a Neural Network mimics the input pattern it was presented with, then that network is said to be autoassociative. For example, if a neural network were presented with the pattern "0110" and the output were also "0110", then that network would be said to be autoassociative. A neural network calculates its output based in the input pattern and the neural network's internal connection weight matrix. The values for these connection weights will determine the output from the neural network, based upon input pattern.

During a pattern matching, segmented characters are taken and mapped as input neuron. All neuron nodes weights, defined as:

i) W j (1), j = 1 ⋯n, are initialized randomly.
W is the number of neurons in the output layer.
ii) K =Maximum (X y), for iteration step y=1...K, get an input vector X k from first recognized stroke
iii) Calculate Distance = Xk k = 1 ⋯n 1 ⋯n refers to neuron nodes for all strokes in the character to match.
*iv) Select the winner output neuron j * with minimum distance (which is more resembling to the stroke of testing)*
*v) Update weights W j (k to neurons j * and its neighborhood*
vii) If pattern is not matching, then take adjacent neuron as desired and goto (iv)
vi) If k has more weights from K go to step (ii).

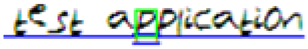
To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.

6. EXPERIMENTAL SETUP

From a hard copy document, image will be extracted for offline character recognition. There are quite a few conventions in determining the input mode of such offline character recognition viz:

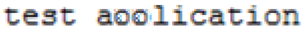
- complete document read / scan
- sequential reading (word / sentence wise) from the document (scan)
- Reading character / word / sentence directly from the digital image of the document

This required offline character recognition from a handwritten, multi-lingual document with mixed-language content.
Consider a scanned text "This is a test scan" which is fed to the system for recognition as follows:



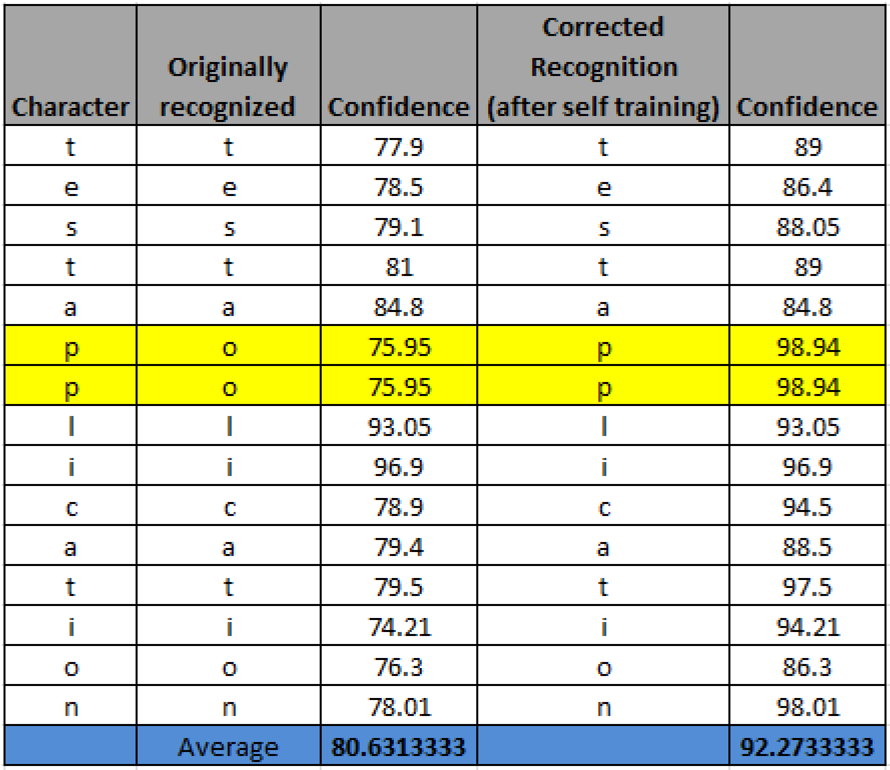
test application

The system first learns the strokes by itself and maps to the character set (alphabet series) it stores in the knowledge base. Hence before training (including pre-processing and noise reduction) it is able to recognize some of the characters like



test aoolication

The confidence of this recognition is about 73.49%. After the character set is manually corrected in the system (re-trained) for recognition, there is a good improvement in the system where it recognized the characters with 91.22% confidence like
as summarized below:



Character	Originally recognized	Confidence	Corrected Recognition (after self training)	Confidence
t	t	77.9	t	89
e	e	78.5	e	86.4
s	s	79.1	s	88.05
t	t	81	t	89
a	a	84.8	a	84.8
p	o	75.95	p	98.94
p	o	75.95	p	98.94
l	l	93.05	l	93.05
i	i	96.9	i	96.9
c	c	78.9	c	94.5
a	a	79.4	a	88.5
t	t	79.5	t	97.5
i	i	74.21	i	94.21
o	o	76.3	o	86.3
n	n	78.01	n	98.01
	Average	80.6313333		92.2733333

TABLE 1: Training Metrics for the Input Source.

<u>Steps involved:</u>

Step 1: Obtain a stroke
Step 2: Normalize stroke
Step 3: Generate Index
Step 4: Obtain a stroke
Step 5: Create index structure
Step 6: Index retrieval
Step 7: Grouping characters
Step 8: Store the character set with index

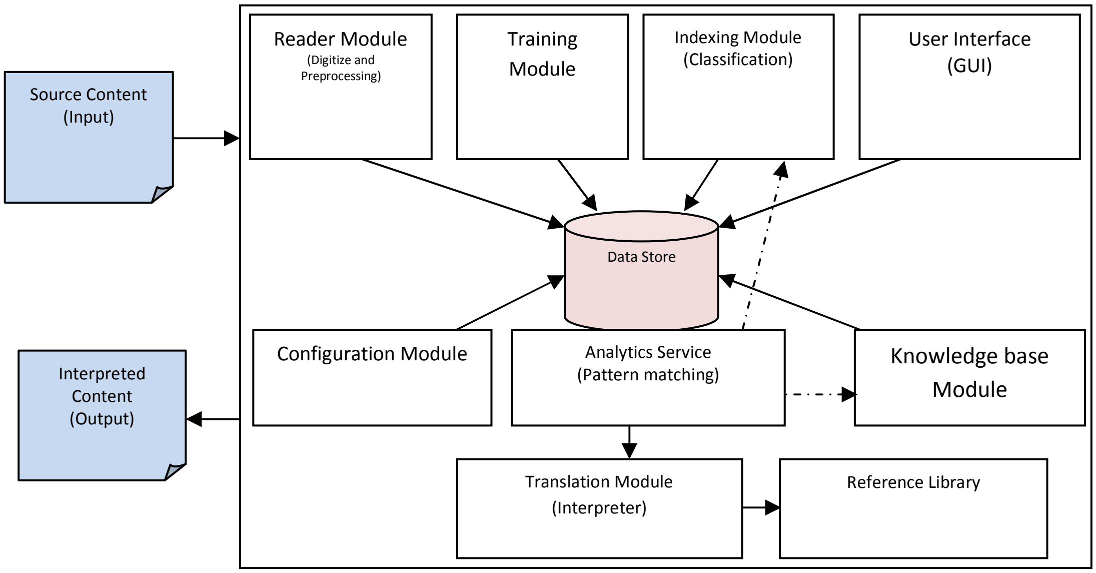


FIGURE 2: System Architecture of Proposed System.

Metrics on individual character recognition **Accuracy Ratio** is shown below:

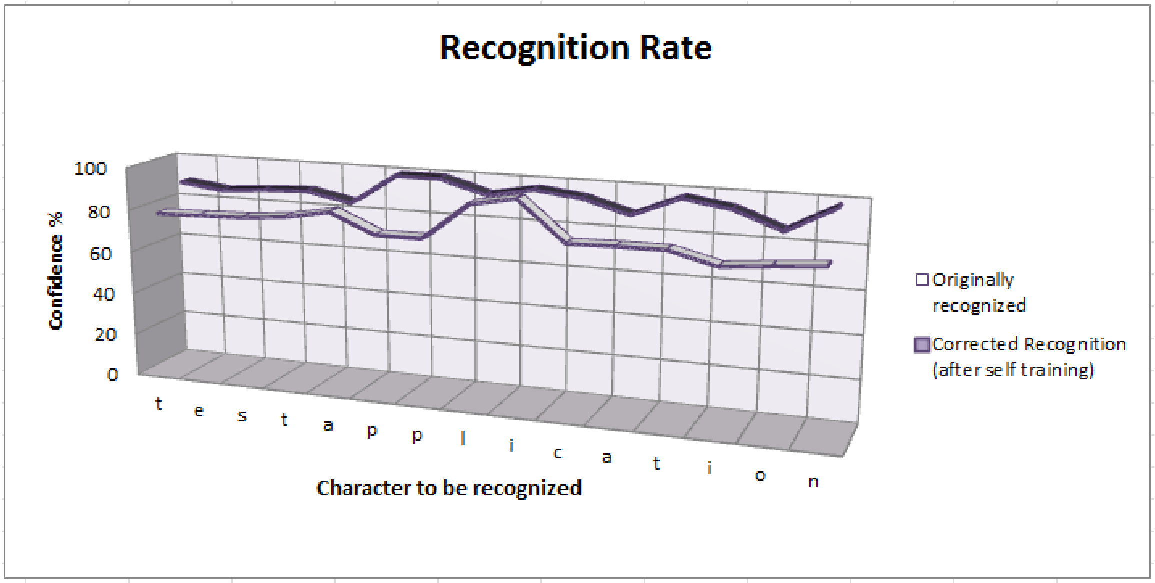
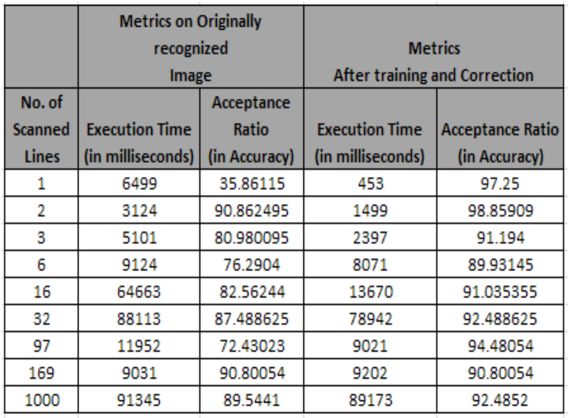


FIGURE 3: Metrics – Acceptance Ratio.

With this setup, when a distributed variance of input text are fed to the system for recognition, there would be interesting statistics based on number of lines to scan and time taken in producing the result co-related with accuracy level of the recognition.

This sample data processing detail is summarized below:



	Metrics on Originally recognized Image		Metrics After training and Correction	
No. of Scanned Lines	Execution Time (in milliseconds)	Acceptance Ratio (in Accuracy)	Execution Time (in milliseconds)	Acceptance Ratio (in Accuracy)
1	6499	35.86115	453	97.25
2	3124	90.862495	1499	98.85909
3	5101	80.980095	2397	91.194
6	9124	76.2904	8071	89.93145
16	64663	82.56244	13670	91.035355
32	88113	87.488625	78942	92.488625
97	11952	72.43023	9021	94.48054
169	9031	90.80054	9202	90.80054
1000	91345	89.5441	89173	92.4852

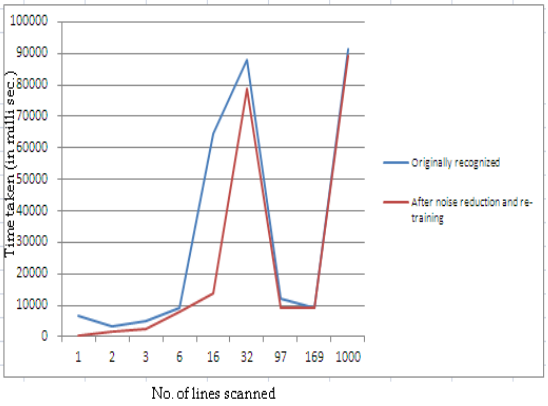
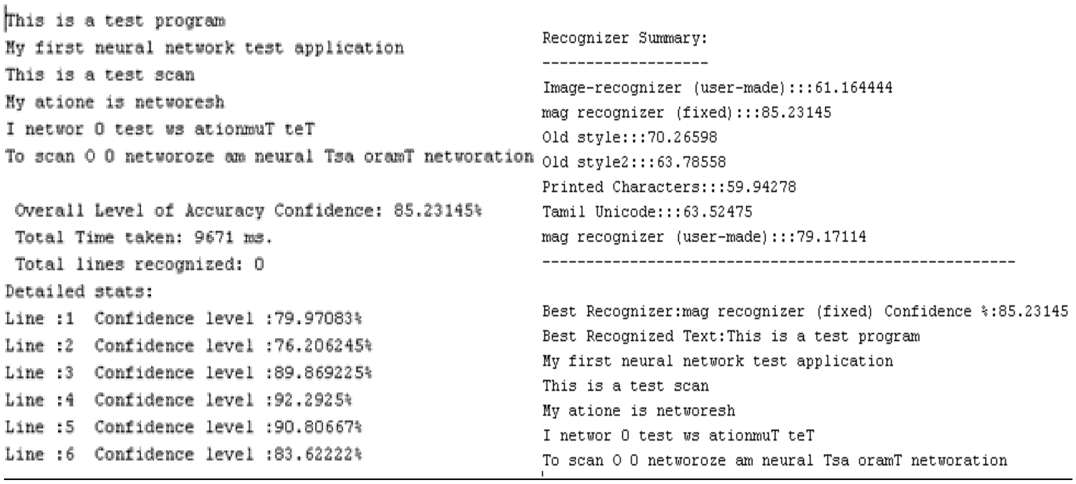


FIGURE 4: Metrics on Execution Time and Acceptance Ratio.

Various users write in varied handwriting styles and in different languages. Each character in the handwritten text may have been written in multiple handwriting styles by different users. It is desirable that a character be correctly recognized in spite of having been written in varied handwriting styles. In addition, a character in one language may be similar, but not same, to a character of a different language and is, thus, prone to be incorrectly recognized.



```
This is a test program
My first neural network test application
This is a test scan
My atione is networesh
I networ O test ws ationmuT teT
To scan O O networoze am neural Tsa oramT networation

 Overall Level of Accuracy Confidence: 85.23145%
 Total Time taken: 9671 ms.
 Total lines recognized: 0
Detailed stats:
Line :1  Confidence level :79.97083%
Line :2  Confidence level :76.206245%
Line :3  Confidence level :89.869225%
Line :4  Confidence level :92.2925%
Line :5  Confidence level :90.80667%
Line :6  Confidence level :83.62222%
```

```
Recognizer Summary:
-------------------
Image-recognizer (user-made):::61.164444
mag recognizer (fixed):::85.23145
Old style:::70.26598
Old style2:::63.78558
Printed Characters:::59.94278
Tamil Unicode:::63.52475
mag recognizer (user-made):::79.17114
-----------------------------------------------------

Best Recognizer:mag recognizer (fixed) Confidence %:85.23145
Best Recognized Text:This is a test program
My first neural network test application
This is a test scan
My atione is networesh
I networ O test ws ationmuT teT
To scan O O networoze am neural Tsa oramT networation
```

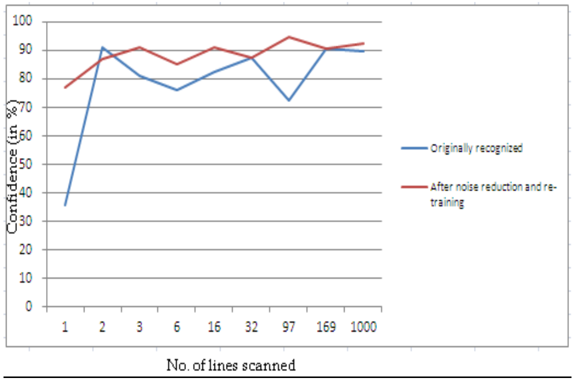


FIGURE 5: Metrics – Confidence Accuracy in Recognition.

Consider a simple example of learning a character ‘A’ based on mapping tables given above. The steps involved in this learning process are explained below:

6.1. Input

From an input source (scanner), the image representing the text may be received by the electronic device by means of scanning a handwritten document.

6.2. Pre-process

A processor of the electronic device may preprocess the received image. The preprocessing may include digitization of the received image.

The preprocessing may further include removing noise such as, but not limited to, salt and pepper noise and Gaussian noise from the digitized image using one or more noise removal techniques known in the art. In addition, the preprocessing may also include making the width of all the characters in the received image uniform by normalizing the width of each portion of the text to a predetermined value of width.

6.3. Normalization

This may include either reducing width of some portions of the handwritten text or increasing their width to the predetermined value of width. The width of a portion may be reduced by converting any undesired black pixels to white if the handwritten text is represented by black pixels.

Once the received image is preprocessed, the received image is segmented into one or more first strokes by the processor.

6.4. Segmentation

On preprocessing the image, the processor of the electronic device may segment the handwritten text in the received image into characters. The processor may distinguish one component of the text from another component based on spacing between the components.

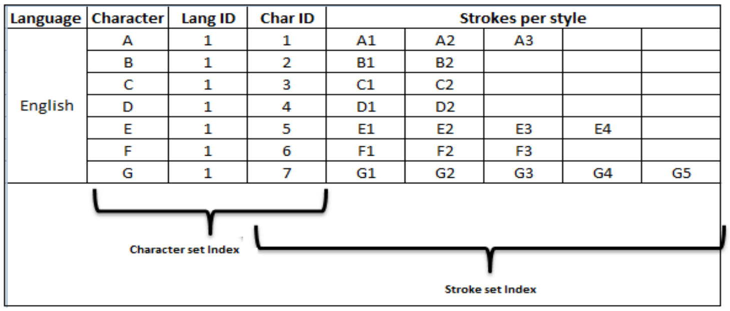
6.5. Strokes Preparation

A sudden change in angle may be considered at a point on a character when two linear or non-linear line segments form an angle at that point that is below a predetermined threshold angle.

For example, if an angle formed by two line segments at a point is below a predetermined threshold angle of 40°, it may be considered as a sudden change in angle. Once all the points representing a sudden change in angle have been identified, the processor may split character at these points into different strokes.

6.6. Stroke Recognition

The processor may scan each of these cells that represent a portion of a stroke sequentially to determine one or more parameters associated with a portion of another stroke represented in that cell.



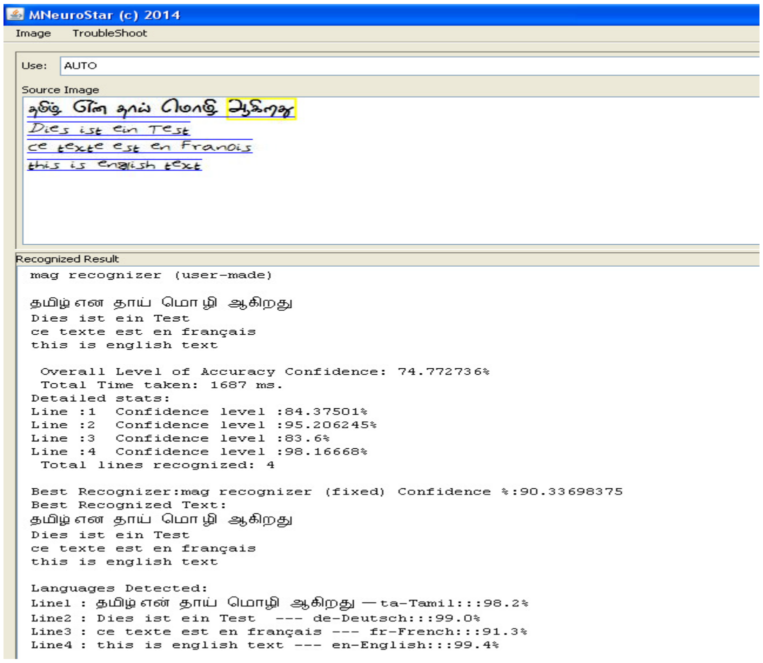
Language	Character	Lang ID	Char ID	Strokes per style				
English	A	1	1	A1	A2	A3		
	B	1	2	B1	B2			
	C	1	3	C1	C2			
	D	1	4	D1	D2			
	E	1	5	E1	E2	E3	E4	
	F	1	6	F1	F2	F3		
	G	1	7	G1	G2	G3	G4	G5

Character set Index
Stroke set Index

FIGURE 6: Indexed List of Strokes for Character Sets In Knowledge Base.

7. MIXED LANGUAGE RECOGNITION

Once the text is recognized based on character sets available in the system, then comes mixed language detection, Offline recognition API like LangDetect or Online detection API from Google can be used. They support various profiles using Unicode based character recognition with confidence level detection as well. This helps in deciding best language possibility based on higher detection confidence.



Use: AUTO

```
mag recognizer (user-made)

தமிழ் என தாய் மொ ழி ஆகிறது
Dies ist ein Test
ce texte est en français
this is english text

 Overall Level of Accuracy Confidence: 74.772736%
 Total Time taken: 1687 ms.
Detailed stats:
Line :1  Confidence level :84.37501%
Line :2  Confidence level :95.206245%
Line :3  Confidence level :83.6%
Line :4  Confidence level :98.16668%
 Total lines recognized: 4

Best Recognizer:mag recognizer (fixed) Confidence %:90.33698375
Best Recognized Text:
தமிழ் என தாய் மொ ழி ஆகிறது
Dies ist ein Test
ce texte est en français
this is english text

Languages Detected:
Line1 : தமிழ் என் தாய் மொழி ஆகிறது — ta-Tamil:::98.2%
Line2 : Dies ist ein Test  --- de-Deutsch:::99.0%
Line3 : ce texte est en français --- fr-French:::91.3%
Line4 : this is english text --- en-English:::99.4%
```

FIGURE 7: Mixed Language Detection.

In Mixed language detection, the idea is to first recognize each line/word of text using all character set available in the knowledge base and then detect the language using language detection API. The system is planned in such a way that the knowledge base will be in synchronous with the language detection profile to have equal or more set of profiles for language detection to enable all possible detections for languages.

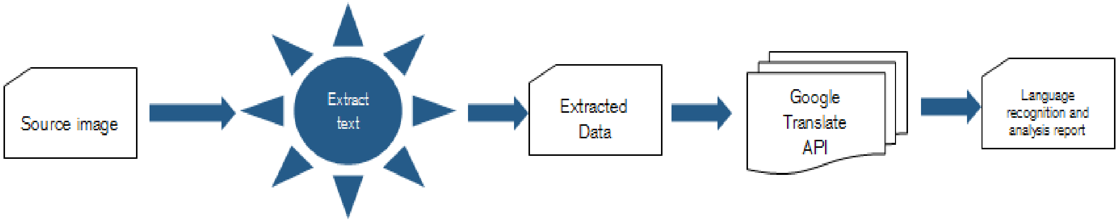
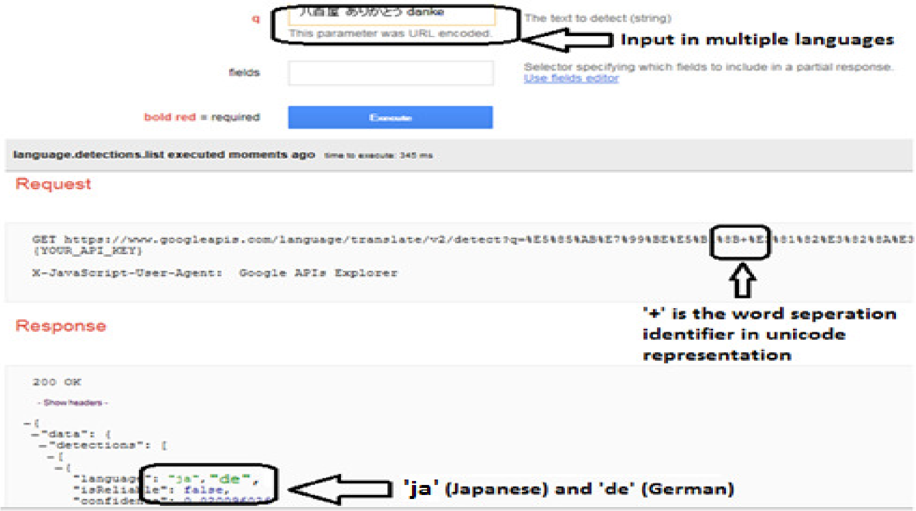


FIGURE 8: Technique in Character Extraction and Language Detection.

8. DETERMINING PROCESSING ACCURACY

When we have a document having multiple languages, then it would be always a tough process to detect languages. It is a tedious and error prone process in automated language detection or in manual process as the translator person need to understand all the languages used in the document.

Though the possibility of such a document is less as there is no real-time usecase available for such a requirement, it is always best to handle all possible alternate situations in usecases to avoid or minimize the mistakes in language detection and improvise the accuracy of the language detection process.



q: 八百屋 ありがとう danke
This parameter was URL encoded.
The text to detect (string)

Input in multiple languages

fields: Selector specifying which fields to include in a partial response.
Use fields editor

language.detections.list executed moments ago time to execute: 345 ms

```
GET https://www.googleapis.com/language/translate/v2/detect?q=%E5%85%AB%E7%99%BE%E5%B%8B+%E%81%82%E3%82%8A%E3
{YOUR_API_KEY}

X-JavaScript-User-Agent:  Google APIs Explorer
```

'+' is the word seperation identifier in unicode representation

```
200 OK

-{
 -"data": {
  -"detections": [
   -[
    -{
      "language": "ja","de",
      "isReliable": false,
      "confidence": 0.0200
```

'ja' (Japanese) and 'de' (German)

FIGURE 9: Sample of Language Detection (Fixed algorithm for Multi-lingual content).

9. FURTHER RESEARCH AND IMPROVEMENT

The idea of this research is to develop a unified system for character recognition and language detection in mixed language content. The idea is being further developed to integrate with online translators like Google Translate API or LangDetect API library to do instant translation as well along with character recognition and language detection. Language detection helps in determining the source of language to translate and this helps in a complete end-to-end processing system for recognition and translation together.

Also, this is a challenging area as it helps a lot of training data and offline dictionary elements to do mixed language content based translation to bring in all content to one single language.

10. RESULT OF EVALUATION

Working on the statistical data points on processing the characters for accuracy, processing time and MSE (mean squared error), posted below a sample test result [19].

In statistics, the mean squared error (MSE) of an estimator is one of many ways to quantify the difference between values implied by an estimator and the true values of the quantity being estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. MSE measures the average of the squares of the "errors." The error is the amount by which the value implied by the estimator differs from the quantity to be estimated. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate.

The processor may also determine a set of first parameters for other strokes such as those shown in figure 2 into which the image of character ‘A’ was segmented by the processor.

This is how self-learning based training is conceptualized based on first stroke identification.

11. ACKNOWLEDGEMENT

This concept is a sub-section of US/India patent filed concept and acknowledged by the patent (Indian Patent Journal Issue 47/2013 dated 22/11/2013). All content and idea are copyrighted.

12. CONCLUSION

The idea explained in this paper relates generally to text recognition, and more particularly to systems and methods for offline character recognition.

Algorithm	Training sets	Time taken in recognition for Test set (in secs)
Supervised training	43 X 10 character sets	21.240
Gabor Filter	43 X 5 character sets	9.8871
Markov Chains and Markov Filter	43 X 8 character sets	64.663
MNeustar (Self-training)	**43 X 1 character sets**	**6.499**

Algorithm	Accuracy % in Test set 1 (English Handwritten)	Accuracy % in Test set 1 (Mixed language content)
Supervised training (Coates et al.)	94.56%	53.2%
Gabor Filter	91.34%	48.2%
Unsupervised training (Neuman et al.)	95.29%	69.19%
Wang et. Al.	98.14%	91.9%
MNeustar (Self-training)	**97.25%**	**93.25%**

TABLE 2: Metrics Showing Benefit of Proposed Algorithm.

In one embodiment, a character recognition method executed on an electronic device is disclosed, the method comprising: receiving, at the electronic device, an image representing a character including one or more first strokes; determining a set of first parameters associated with each of the one or more first strokes; comparing, for each of the one or more first strokes, the associated set of first parameters with a plurality of stored sets of second parameters, wherein each of the plurality of stored second strokes is associated with a stored set of second parameters; identifying a second stroke, from among the plurality of stored second strokes, corresponding to each of the one or more first strokes based on the comparison; and identifying the character based on the identified one or more second strokes.

13. References

[1] Wai Kin Kong, David Zhang, Wenxin Li, Palmprint feature extraction using 2-D Gabor flters, The Journal of the Pattern Recognition Society (Elsevier) Pattern Recognition 36 (2003) 2339 - 2347.

[2] Daming Shi, Robert I. Damper And Steve R. Gunn, Offline Handwritten Chinese Character Recognition by Radical Decomposition, ACM Transactions on Asian Language Information Processing, Vol. 2, No. 1, March 2003, Pages 27-48.

[3] Anita Pal, Dayashankar Singh, Handwritten English Character Recognition Using Neural Network, International Journal of Computer Science & CommunicationVol. 1, No. 2, July-December 2010, pp. 141-144.

[4] R.Jagadeesh Kannan And R.Prabhakar, Off-Line Cursive Handwritten Tamil Character Recognition, WSEAS Transactions On Signal Processing, Issue 6, Volume 4, June 2008, ISSN: 1790-5052 Pages: 351-360.

[5] Lubna Badri, Development of Neural Networks for Noise Reduction, The International Arab Journal of Information Technology, Vol. 7, No. 3, July 2010 Pages: 289-294

[6] Mansi Shah And Gordhan B Jethava, A Literature Review On Hand Written Character Recognition, Indian Streams Research Journal, Vol -3 , ISSUE 2, March.2013, ISSN:-2230-7850.

[7] Zhiyi Zhang, Lianwen Jin, Kai Ding, Xue Gao, Character-SIFT: a novel feature for offline handwritten Chinese character recognition, 10th International Conference on Document Analysis and Recognition, 2009 Pages: 763-767.

[8] Li Fuliang, Gao Shuangxi, Character Recognition System Based on Back-propagation Neural Network, 2010 International Conference on Machine Vision and Human-machine Interface Pages: 393-396.

[9] Dr.J.Venkatesh and C. Sureshkumar, Tamil Handwritten Character Recognition Using Kohonon's Self Organizing Map, IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.12, December 2009 Pages: 156-161.

[10] L. D. Jackel, C. E. Stenard, H. S. Baird, B. Boser, J. Bromley, C. J. C. Burges, J. S. Denker, H. P. Graf, D. Henderson, R. E. Howard,W. Hubbard, Y. leCun, 0. Matan, E. Pednault, W. Satteriield,E. Sickinger, and T. Thompson, A Neural Network Approach to Handprint Character Recognition, IEEE CH2961-1/91/0000/0472 2001 Pages: 472-475.

[11] Seong-Whan Lee, Young- Jaon Kim, A New Type of Recurrent Neural Network for Handwritten Character Recognition, IEEE 0-8186-7128-9/95 2005 Pages: 38-41.

[12] Ishwarya .M.V, R. Jagadeesh Kannan, An Improved Online Tamil Character Recognition Using Neural Networks, 2010 International Conference on Advances in Computer Engineering IEEE 978-0-7695-4058 Pages: 284-288.

[13] G. Tambouratzis, Applying Logic Neural Networks to Hand-written Character Recognition Tasks, IEEE 0-8186-7686-8/9 10996 Pages : 268-271.

[14] Anil K.Jain, Jianchang Mao, K.M.Mohiuddin, Artificial Neural Networs : A Tutorial, IEEE 0018-9162/96 March 1996 Pages: 31-44.

[15] Neural Networks, Fuzzy Logic and Genetic Algorithms – Sythethis and Applications by S.Rajasekaran and G.A.Vijayalakshmi Pai from Eastern Economy Edition Page-31-33.

[16] LI Guo-hong,SHI Peng-fei, An approach to offline handwritten Chinese character recognition based on segment evaluation of adaptive duration, Journal of Zhejiang University Science ISSN 1009-3095 2004 5(11):Pages: 1392-1397.

[17] Magesh Kasthuri, Dr. V.Shanthi, Noise Reduction and Pre-processing techniques in Handwritten Character Recognition using Neural Networks, TECHNIA International Journal of Computing Science and Communication Technologies, VOL.6 NO. 2, January. 2014 (ISSN 0974-3375) Pages: 940-947.

[18] Magesh Kasthuri, Dr.V.Shanthi Self-training Method using First strokes in Handwritten Character Recognition International Journal of Scientific Research, Vol.III, Issue. V, May 2014, ISSN No. - 2277-8179 Pages: 73-77.

[19] Magesh Kasthuri, Dr.V.Shanthi Pre - processing and Self training techniques in Handwritten Character Recognition Indian Journal of Applied Research, Vol.IV, Issue. IV April 2014 ISSN - 2249-555X – Pages: 189-193.