International Journal of Computational and Experimental Science and ENgineering
(IJCESEN)
Vol. 11-No.1 (2025) pp. 1377-1388
http://www.ijcesen.com
ISSN: 2149-9144

Research Article

A Graph-Based and Pattern Classification Approach for Kannada Handwritten Text Recognition Under Struck-Out Conditions

H. K. Bhargav[1], Ambresh Bhadrashetty[2], K. Neelashetty[3], V. B. Murali Krishna[4], G. Manohar Bali [5*]

[1]Director, Industry Institute Interaction Cell (IIIC) & Associate Professor in Department of Computer Science and Engineering, Shridevi Institute of Engineering and Technology, Tumakuru 572 106, Karnataka, India.
Email: bhargavwin@gmail.com - **ORCID:** 0000-0002-4922-3753

[2] Department of Computer Science and Engineering (MCA), Visvesvaraya Technological University, Centre for PG Studies, Kalaburagi 585105, Karnataka, India.
Email: ambresh.bhadrashetty@gmail.com - **ORCID**: 0000-0003-4443-9018

[3] Department of Electrical and Electronics Engineering, Guru Nanak Dev Engineering College, Bidar 585 403, Karnataka, India.
Email: neelshettyk@gmail.com – **ORCID:** 0000-0003-2822-8747

[4] Department of Electrical Engineering, National Institute of Technology Andhra Pradesh, Tadepalligudem 534 101, India.
Email: muralikrishna.cuk@gmail.com - **ORCID:** 0000-0003-0643-8217

[5] School of Engineering, Department of Computer Science and Engineering, Central University of Karnataka, Kadaganchi 585 367, Karnataka, India.
* **Corresponding Author Email:** manoharbali.cse@gmail.com - **ORCID:** 0009-0008-7302-4528

Article Info:

DOI: 10.22399/ijcesen.1021
Received : 05 January 2025
Accepted : 8 February 2025

Keywords :

Crossed-out text,
Kannada language,
Optical character recognition (OCR),
Machine learning.

Abstract:

This research focuses on the processing and identification of handwritten Kannada text, particularly under struck-out conditions. The database considered in this study comprises handwritten data. When such a database is processed using optical character recognition (OCR)-based digital systems, the output may often be in an unrecognizable format. To address this issue, a model has been developed incorporating pattern classification and a graph-based method for text identification. For pattern classification, feature extraction is performed using two different classes with a support vector machines (SVMs) classifier. In the graph-based approach, struck-out strokes are analyzed using the shortest path algorithm. To handle zigzag or wavy struck-out Kannada text, all possible paths of the strike-out strokes are identified, and suitable features are extracted for further processing. The synthesized/recovered text is processed using an inpainting cleaning method to ensure text recovery. The proposed methodology has been tested on both trained and untrained datasets of Kannada script. Performance evaluation was conducted using three parameters: precision, F1 score, and accuracy.

1. Introduction

In recent decades, the transcription of offline handwriting in digital document analysis has played a crucial role. Several researchers have focused on improving recognition patterns and enhancing the accuracy of reading handwritten documents. In the initial stages of handwriting recognition research, isolated characters were predominantly considered for recognition. However, in practical scenarios, the data often contains a mix of single and multiple characters. Additionally, errors such as strikeouts of words in handwritten text are commonly observed. Consequently, after word recognition, the detection of struck-out text becomes a critical step in handwriting analysis. Deep neural network techniques offer promising solutions for addressing these challenges [1-3].

The Kannada script consists of 49 letters, including vowels, consonants, and syllable consonants, and bears a strong resemblance to other Brahmic scripts. This paper explores the characteristics of Kannada script and investigates methodologies to handle struck-out words effectively in the context of handwriting analysis [2].

This paper focuses on the analysis of struck-out words in Kannada script. A comparison is also made between Kannada and English characters and words. Kannada script, referred to as an abugida of the Brahmic family, is one of the four major Dravidian languages of South India. Kannada is predominantly spoken in Karnataka and encompasses several dialects and minor languages such as Kodava, Beary, Konkani, and Tulu, all of which share the Kannada script. The script has a rich history spanning thousands of years and exhibits significant mutual intelligibility with Telugu script. This paper aims to pre-process handwritten text using OCR and identify strike-out strokes in Kannada literature. Different types of strike-out strokes are illustrated in Figure 1. As shown in Figure 1, there are various strike-out stroke structures, including single, multiple, slanted, crossed, zigzag, and wavy strokes. Among these, the most observed types are single strokes, multiple strokes, slanted strokes, crossed strokes, zigzag strokes, and wavy strokes [4-8].

A horizontal line drawn over a word represents a single strike-out stroke (SSS). Alternatively, a zigzag line can be used to obscure the word. In some cases, a zigzag line may also exhibit wavy characteristics, partially concealing the written text. Slanted lines are another common form of strike-out strokes [9-13]. These can consist of either a single slant line or multiple slant lines drawn over the word, serving the same purpose [14-16]. The angle of the slanted lines typically ranges from 10 to 60 degrees. Additionally, slanted lines may extend both above and below the word, further complicating the recognition process [3].

In Figure 2, a complete paragraph of text is shown to be clearly struck out. In handwriting analysis, consecutive words, a complete paragraph, or even an entire page can be struck out. This study aims to provide a comprehensive solution for categorizing handwritten scripts from Kannada and English alphabets, identifying struck-out characters and words in the text. The process is divided into three subtasks.

1. Identifying and separating normal text from struck-out text.
2. Performing stroke identification.
3. Removing the strokes and producing text suitable for OCR analysis.

Depending on the requirements, the user may skip the third task and focus solely on identifying the struck-out text [4]. Character recognition in this paper is implemented using a randomized algorithm. This method is preferred as it minimizes the

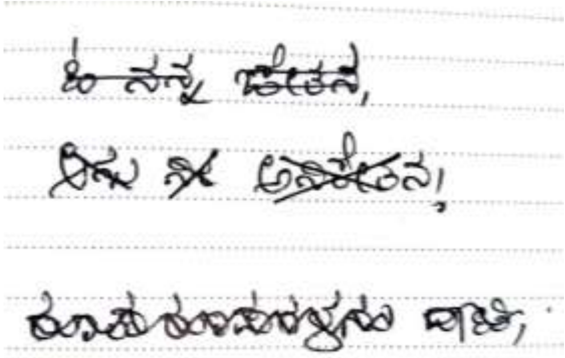


Figure 1. *Various forms of strikeout strokes on Kannada letters/words.*

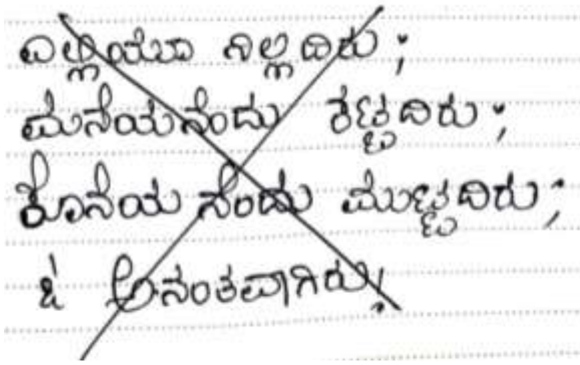


Figure 2. *The strikeout strokes of complete paragraph and the page.*

Thinned Binary Images (TBI) from the length of the Freeman Chain Code (FCC) [17-24]. The starting point plays a critical role in determining the FCC length. One challenge with FCC is that the representation of characters depends on the length of the FCC. During FCC generation, each pixel must be traversed, which involves several branches, and revisiting the same node can become tedious. To address this issue, heuristics are employed to develop an accurate FCC, which is then used to represent characters. Each character is clearly distinguished at every classification stage. For character recognition, the SVM (Support Vector Machine) algorithm is employed [5,11,12].

SVM is one of the most widely used supervised learning algorithms, particularly for classification problems. It is also applicable to regression tasks. SVM works by determining a decision boundary, often referred to as the best line, that separates data points in a dimensional space. This boundary separates the data into distinct classes, aiding in the recognition of scanned characters. The hyperplane in SVM represents the optimal decision boundary.

There are two types of SVMs: linear and non-linear.

- *Linear SVM:* Used for linearly separable data, where a single straight line can classify the data into two distinct classes. The classification

performed by this algorithm is referred to as a linear SVM classifier.
- *Non-linear SVM:* Used for non-linearly separable data, where a single straight line cannot classify the data. Instead, multiple straight lines are utilized to separate the data, leading to what is known as a non-linear SVM classifier.

The organization of the paper is as follows: Section 2 covers the existing systems, including prior research and their results regarding the accuracy of character identification. Section 3 discusses the proposed block diagram, and the key technologies used in this study.

2. Material and Methods

To understand the recognition system for handwritten characters, the combination of artificial intelligence, pattern recognition, and computer vision concepts is crucial. A simple computer application capable of performing handwritten character recognition can detect and interpret characters from papers, images, or any other electronic format that can be machine-encoded. Handwriting can be scanned through intelligent word recognition, optical scanning, or directly written using a digital pen on a screen.

In some automated systems, characters written on a screen are processed immediately as the pen interacts with the surface, providing cues to the recognition algorithm. Handwriting recognition is categorized into two types: offline recognition and online recognition.

- *Offline Recognition:* Involves extracting letter codes from static, pre-stored images. The process converts handwritten documents into a digital format using algorithms that analyze static representations of handwritten images.
- *Online Recognition:* Uses specialized screens where characters are written with a pen, capturing the movement of the pen tip to recognize the word or character being written.

Various techniques, such as neural networks, machine learning algorithms, Hidden Markov Models (HMM), and Support Vector Machines are employed to ensure accurate character detection and recognition [9-12].

In this study, an untrained database was used, comprising samples collected from Kannada-medium school children aged 10 years ± 3 years. The participants were native Kannada speakers with at least four years of experience in writing Kannada. Twenty students were randomly selected, informed about the study, and asked to write text from the poem Vishwamanava Sandesha by the renowned Kannada poet Kuvempu on blank sheets. The handwritten samples were digitized using an appropriate scanner and processed using Python with supportive library functions [12,13].

2.1 Pre-Processing and Segmentation Techniques

The author of ref [14] classifieds the category of handwriting recognition system into 2 types. The first one is visual appearance-based technique and the second one as structure-based techniques. This paper gives a brief survey on different scripts that are being used in the methodology and these categories. The author also proposes the model to understand and to identify the text in videos and in the online data. The author gives a special note saying that the research area is still a thin Line, and many particularities have to be developed based on handwritten documents. The research as discussed in [15-19] discuss about optical character recognition system that is already present and associating the symbol identities with the Individual characters. The author explains the functionality of script recogniser using the language of Urdu and English. The author discusses regarding several languages in Brahmi script. The script recognition methodologies that are currently used in machine learning environment are also mentioned. The Spitz method that is used in this paper is described with a clean block diagram and different languages that can be recognised. Using this system the author was able to recognise English, French, German. And many other characteristics shape words, and optical density distributed the languages like Chinese, Japanese and Koreans were also identified. The language recognition was based on pen position and the samples collected at the rate of 132 samples per second. The algorithm also had an impact of strokes and resemblance. To smoothen the image, Gaussian filter was used. The X axis projection of the text is used to perform the word segment analysis. But the segmentation will not work whenever the different regions of handwritten text with different scripts are used [16].

2.2 Feature Extraction, Classification and Recognition Techniques

The system of multistage recognition scheme is is discussed in the literature [20-26]. The 3 multilayers are cascaded as multilayer perceptron in the first stage. The decision is not in the possible class. Then the images cannot reach the next stage of multilayer perceptron. Another MLP is used to classify the image, which is fed, to the second stage [28-31]. 20 class handwritten Devanagari data is taken for the classification accuracy. The accuracy was found to be 70.85% and the English mixed Devanagari Script had a classification of 65.02%. [17].

The architecture use is termed as connectionist architecture. The classifier is a multi-classifier which consisted of the following key concepts.

The classifier is a multi-classifier which consisted of the following key concepts.

1. Kohonene Self Organizing Map at the lowest layer.
2. A single-layer super-structure laid on Kohonens Net.
3. Multilayer perceptron's (MLP) based classifier for segment features.
4. Meta-pi combining net for integration.

For the offline word recognition system, the author uses hidden Markov's model in paper [12]. There are some simple superimposed strokes on the neat handwriting of English word. These imposed strokes are simulated using line trajectory and wave trajectory techniques. They are not the strike out penned by the natural handwriting. So, no attempt for identification or cleaning of such simulation is done [18].

The authors [19] discuss feature extraction and classification techniques essential for recognizing handwritten Kannada characters, emphasizing the role of machine learning algorithms in training extracted features for both online and offline handwritten Kannada character recognition, addressing existing challenges in the process. In [20], the authors employ zonal, pattern, and gradient feature extraction techniques, followed by significant feature selection using ISSA. The BMCNN classifier is then utilized for recognizing handwritten Kannada characters, demonstrating improved performance compared to existing methods. The work [21] discusses feature extraction using geometric methods and character recognition through a LSTM network, addressing challenges in handwritten Kannada scripts, including varying writing styles and the lack of large training datasets. The accuracy of the classification technique was improved to 89.68% for Devanagari numerical. For the offline word recognition system, the author uses hidden Markov's model in paper [12]. There are some simple superimposed strokes on the neat handwriting of English word. These imposed strokes are simulated using line trajectory and wave trajectory techniques.

3. Proposed System

The proposed system classifies strikeouts into three major types:

1. *Successive Multi-Line Strike-Out:* Occurs when multiple lines are consecutively struck out.
2. *Single Word Strike-Out:* Refers to individual words that are struck out.
3. *Successive Multi-Word Strike-Out:* Refers to multiple consecutive words that are struck out.

The workflow of this system is illustrated in Figure 3 [19].

To identify text lines in the document images, an off-the-shelf algorithm is employed. This algorithm effectively detects the presence of strikeouts within the document. Each text line is analyzed, and the Current Component (CC) is examined. At this stage, smaller components such as colons, dashes, and dots are also identified, though these are excluded from further processing [20]. An advantage of strike-out words is that they are fewer in number compared to other words in the document. The height of a word is estimated relative to the Average Height (Hav) of components within the boundary box. The algorithm also computes the Standard Deviation (Hsd) of the boundary box height. Subsequently, the length of each boundary box is calculated, which helps refine the recognition process [21]. For improved outcomes, the proposed method uses an integer multiplier α\alphaα, with α\alphaα set to 8 in this study. When the height of a connected component exceeds this range, it falls into the second category (Case B) [22]. The first step in the classification process is to distinguish between two classes:

1. Non-Strike-Out Class (Class 1): Words or components that are not struck out.
2. Strike-Out Class (Class 2): Words or components identified as struck out.

The SVM classifier was chosen due to its powerful capabilities in separating these two classes effectively. Its performance has been demonstrated to be robust and comparable with other classifiers in similar applications [23-25]. A unidirectional graph is employed to represent the skeleton structure of the strike-out data, as depicted in Figure 3. To identify the strike-out strokes, the system determines the near-horizontal shortest path in the image, spanning from the leftmost node to the rightmost node. Machine learning is used in different application [31-34]. The shortest path is used to trace the strike-out in the data [35-37].

However, determining the shortest path between numerous components in a document can be computationally expensive. To optimize this process, a feature-based SVM classifier is applied, enabling faster and more accurate identification of strike-out elements. A detailed explanation of this methodology is provided in the following sections.

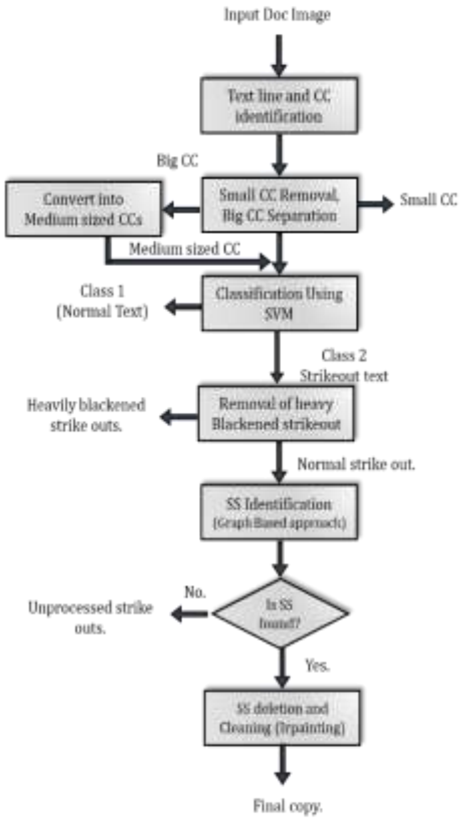


Figure 3. *Workflow of proposed approach.*

3.1 Finding the strike out component:

This section highlights the novel feature of the proposed methodology for detecting strike-out components. Additionally, the classifier employed for the separation of the two classes (strike-out and non-strike-out) is presented and explained in detail.
Feature Selection:
The primary parameters used in this methodology include:

1. *Height of the Component (Hcc):* The height of the connected component within its boundary box.
2. *Width of the Component (Wcc):* The width of the connected component in the boundary box.
3. *Elongation (Ecc):* A parameter that captures the degree of elongation of the component.

All these parameters are linked to the connection line, forming the foundational elements of the strike-out detection system (1).

$$E_{cc} = \frac{min\{H_{cc}, W_{cc}\}}{max\{H_{cc}, W_{cc}\}} \quad (1)$$

Normalization Process:
To ensure accurate and consistent processing, all feature factors are normalized. The normalization steps include:

1. Normalizing Branch Points: Standardizing the connection points along the stroke branches.
2. Normalizing Weighted Branch Points: Adjusting the weights of the branch points for uniform distribution.
3. Normalizing X-Branch Points: Aligning branch points along the X-axis for consistency.
4. Normalizing Black Pixel Density: Measuring the density of black pixels within the component and normalizing this value.
5. Standard Deviation of Density: Normalizing the density variation to account for irregularities.
6. Normalizing the Number of Holes: Counting and normalizing the holes present in the image component.
7. Normalizing Hole Pairs: Identifying and normalizing pairs of holes connected by a common line.

This normalization ensures that all features are processed uniformly, thereby enhancing the robustness of the strike-out detection mechanism.

3.2 SVM Classifier:

Support Vector Machines have gained significant popularity in character recognition systems due to their robust classification capabilities and adaptability across various document image processing applications. SVM, when combined with Radial Basis Function (RBF), enhances the classifier's performance in complex multi-layer perceptron neural networks, making it an unparalleled tool in many scenarios.
For instance, studies have shown that modified quadratic discriminant functions, in conjunction with SVM, perform well in the classification of scripts such as Devanagari. Similarly, this paper employs SVM with RBF kernels to effectively address the challenges of recognizing strike-out texts in both Kannada and English scripts.
The choice of SVM with RBF is motivated by its ability to handle non-linear decision boundaries and its efficiency in high-dimensional spaces. The kernel transforms the input data into a higher-dimensional space where a linear hyperplane can separate classes with maximum margin [1-12].
The workflow of the SVM classifier, as applied in this methodology, is illustrated in Figure 4. The process begins with feature extraction and normalization, ensuring consistent input to the classifier. Features such as elongation, component height, and width are normalized and fed into the classifier. The SVM classifier is then trained to differentiate between two classes—strike-out text and non-strike-out text.
Key Advantages of Using SVM with RBF:

i. Accurate Classification: The RBF kernel allows SVM to model complex, non-linear relationships in the data, ensuring high accuracy in separating strike-out text from regular text.
ii. Robustness: SVM with RBF is less prone to overfitting, even in scenarios with limited training data.
iii. Generalizability: The classifier performs consistently across different datasets, making it ideal for bilingual text processing (Kannada and English).

This approach ensures reliable detection and classification of strike-out text components, paving the way for further processing steps like text cleaning and optical character recognition.

4. Methodology

This section outlines the process of identifying strike-out strokes using a graph-based approach. The methodology employs a systematic procedure to extract the skeletal structure of the text component, filter unnecessary branches, and identify the strike-out stroke effectively.

4.1 Identification of Strike-out Stroke using Graph-based Method

To identify strike-out strokes, the connected components in the graph-based method are represented as SS components. The following steps outline the process:

1. Skeleton Generation:

Using the second thinning algorithm, a skeleton is generated for the text component image (I), denoted as Isk. This skeleton provides a simplified, one-pixel-wide representation of the original text component.

2. Branch Removal:

Any superfluous branches within the skeleton are removed by eliminating small edges. This ensures a clean skeletal structure for further analysis.

3. Terminal Pixel Identification:

A terminal pixel is defined as a skeletal pixel with exactly one neighboring pixel. This is often located at the end of a branch. The neighboring pixel of a terminal pixel that has three or more branches is referred to as a junction pixel.

4. Direct Path Determination:

A direct connection path is established between a junction pixel and its corresponding terminal pixel. This path represents the skeletal segment connecting the two points.

5. Path Deletion:

If the length of the direct path between the junction pixel and the terminal pixel is less than or equal to half the average stroke width (STSTST), the path, along with the terminal pixel, is deleted. This step ensures that only significant skeletal structures are retained.

Skeletal Structure Visualization:

Figure 4 illustrates the skeletal structure generated during the process. The image demonstrates:

- Terminal pixels at the ends of branches.
- Junction pixels at points where three or more branches meet.
- Direct paths connecting terminal and junction pixels.

This graph-based approach simplifies the complexity of strike-out identification by reducing the text component to its essential structure, enabling efficient detection and processing of strike-out strokes.

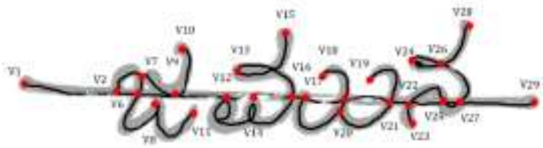


Figure 4: *Skeleton image which is divided into 3 regions: The left region, right region and the mid region. The connection between V1 one and V 28 is showing the SS.*

Apart from the thinning algorithm, good thinning algorithm can also be employed to generate the skeleton structure. The skeleton shape becomes almost identical once the superior branch is burning about the Algorithm. This process is carried out in the good algorithm. One of the main criteria in handwritten text is strokes are they contain small with when compared to length. That is, the normal strokes are highly elongated. So, it can be concluded that the skeleton made by the good approach is merely free from superior branching. The representation of skeleton is done using unit directed graph and it is denoted as (2).

$$\text{Skeleton pair } G = (V, E) \qquad (2)$$

In (2), here in the above equation V is a set of node and E is the set of edges.

4.2 Identification of nodes (V)

The set of nodes or vertices (V) are the terminal pixels or the junction pixels of I_{SK}. All these belongs to the graph G. The notes are termed to be very closely located if they are having a mutual distance less than T_K number of pixels. Only one node is selected such that it has only right movement or left movement during their path estimation. By ceiling the function of half of the S_T, the T_K value is calculated. In this, the average stroke of width I is considered as S_T. Putting everything together, we get: Tk = dST /2e. Image resolution will vary the value of T_K. As the image resolution increases or decreases, the stroke width also varies accordingly, as the stroke width is directly proportional to the image resolution. The notes are commonly indexed from column wise, from left moving from top to bottom of a column.

4.3 Identification of edges (E)

Between the node pixels there exists an edge which is represented as e_{ij}. Assuming that, (vi and (vj | vi, vj ∈ V). They are interconnected by using non node object pixel. There can be 2 or multiple edges between 2 nodes, this is also feasible prompt possibility in the graph. These edges are indicated as e_{ij1}, e_{ij2}, etc. If there exists any self-loop at node I, then they are represented using e_{iL}.

The edges (e_{ij}) are assigned with weight (w_{ij}), by considering the diagonal moves (N_d)and also the horizontal and vertical moves (N_{hv}). The weight is now given by (3).

$$\omega_{ij} = \omega_d N_d + \omega_{hv} N_{hv} \quad (3)$$

4.4 Strike-out stroke as shortest path

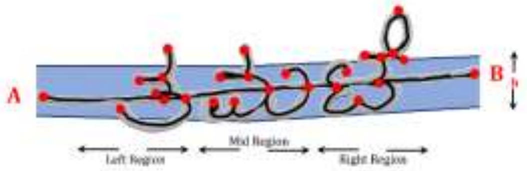
In the earlier step, it is considered to have a reasonable straight SS in the image. In the graph for, the shortest path between left and right node indicates the reasonable straight strike out. In this paper, the approach is based on 2 main observations. The first one is to Observe and horizontal line which covers the width of the component. The second one is to find reasonably straight and continuous line which is unbroken in nature.

4.5 Speed up approach

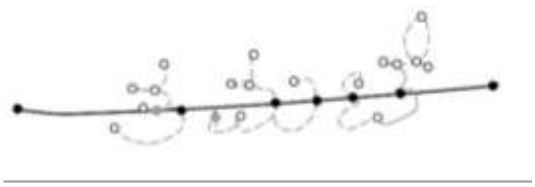
There are 4 steps to analyse the strike outs in speed up approach.

i) The first step is to divide the boundary box into 3 equal components. The 3 equal components are named as a right region, left region and the mid region. In the figure M normally, the strike out is from left to right. And hence we can avoid the middle region nodes as they do not participate in the shortest path.

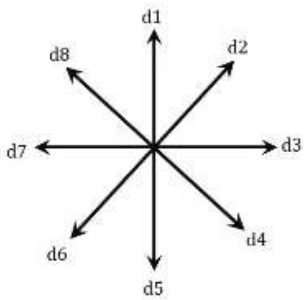
ii) If there are any self-loop in the image, they are deleted. The shortest edge is alone considered whenever there are multiple edges between 2 neighbouring notes. And hence the shortest path is calculated.



(a)



(b)



(c)

Figure 5. *(a) Path from A to B, with the band considered with Height 'h' (b) Generation of a shortest path using band. (c) 8- neighbor traversal direction.*

iii) Considering the shortest path is reasonably street if the 2 nodes are identified between the shortest path are v_i and v_j, and the line v_iv_j joins them both. Then a thick band is created around the line with the thickness of h. The H denotes the tolerance in the deviation from the straightness of shortest path. Hence the head should be selected such a way that it is half the busy text region height.

In the figure 5 (a) skeleton structure of node edge graph is shown. The black notes and the grey notes are representing the path from A to B. The white notes are outside the band. Only the black nodes contribute to the final shortest path generation.

iv) In the observation made as there are no retrograde motion whenever there is a striking from left to right direction. So, the backtracking of the path is not allowed. This helps in the reduction in number of parts. Hence the movement of the analysis can be done in only One Direction as d1, d2, d3, d4 and D5. There are 6 terminal nodes as, v1, v2, v3, v4, v5 etc.

4.6 Shortest path detection

Let the left region has a terminal node as, V_{L1} and the right region have a terminal node as V_{R1}. It is considered such that all the notes have all the possible pixels as a neighbouring object. The shortest path is calculated from left to right, using the Dijkstra's algorithm. Using this algorithm few of the shortest paths are identified. Among them the shortest one is taken.

4.7 Recognising a wavy stroke or a zigzag stroke

One of the difficult and challenging situations in the strikes stroke recognition is when it encounters Azad stroke. The zigzag strokes or the wavy strokes are shown in Figure 4.

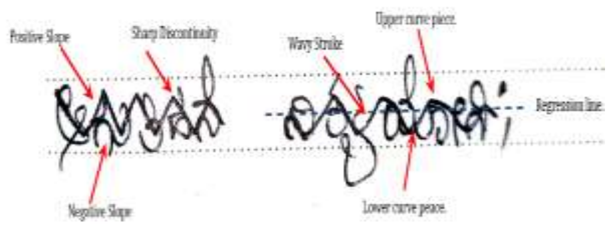


Figure 6. *Processing of wavy strike out and zigzag strike out. The strike out stroke is shown in red colour.*

In this case we can't inculcate the shortest path scheme as the path is neither straight nor short. But there are different parts in the image. Different paths can also be obtained using graph-based method. Using the algorithms we can verify if the path satisfies the property of zigzag stroke. After reasonable observations made some of the examples, some conclusions can be done. The conclusions are such as strike out is done only on the words and it is not more than 2 characters, These zigzag strokes normally covers the entire character of the world, When the average character height is considered, these exact strokes will lie in the middle of the character height, The characteristics of zigzag stroke is that they have positive slope, negative slope and the sharp slope discontinuity, The last observation is the stroke is normally linear between 2 consecutive slopes and having a discontinuity point at the middle. The detailed issues in the example given in Figure 6. To speed up the approach, the number of parts is reduced. The distance between the path from left to right region notes are calculated. The sharp slope and the discontinuity are detected using edge detection technique. The zigzag stroke is confirmed whenever it finds positive slope and a negative slope.

4.8 Recognizing non-horizontal strikeouts

Another example of striking out the words is a slant strike, which is shown in Figure 5. The slant strike may be from left to right or right to left. Based on the observation made, the strike out can be having the following characteristic: It can have a cross mark over the word, it can be a positive slant over the word or a negative slant over the word. If it is a positive slant, it starts from the top right region to bottom left region. If it is a negative slant, it starts from top left region and ends in bottom right region. In the skeletal graph, it is considered only the left to right direction path. So, the connected component of the graph.

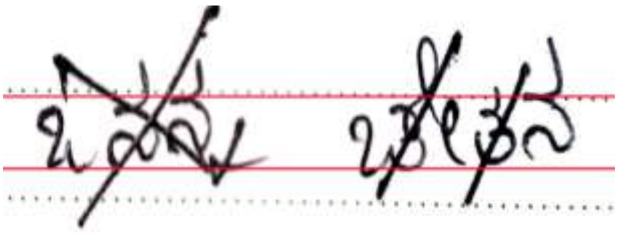


Figure 7. *Direction of near vertical and strike outs.*

As shown in the above figure 7, the slope of summer strike outs is nearly vertical. These are used in the detection of single characters. If the strike out is in this manner, then the height of the strike out is larger than the word component. This can be used to find out the SS easily in long text. [28]

4.9 Handling multiple strikeouts and script-specific strokes

Another unique example of SSs in writing can be seeing in figure 8. The similar procedure to find the strike out word as mentioned in the above can be used here. If the SSs are well spaced, then the detection is very easy. If the SSs are untouched thing, then it is easier to identify the strike out word. At the same time, if there are 3 or more SS drawn on the single word than the identification becomes erroneous. Especially whenever the strokes are very close to each other, and they touch each other. In this paper, only 2 strokes are considered.

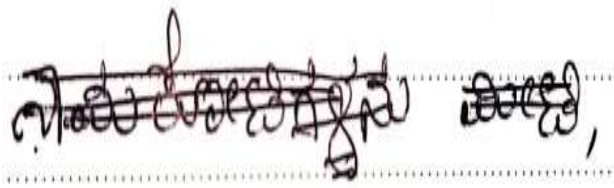
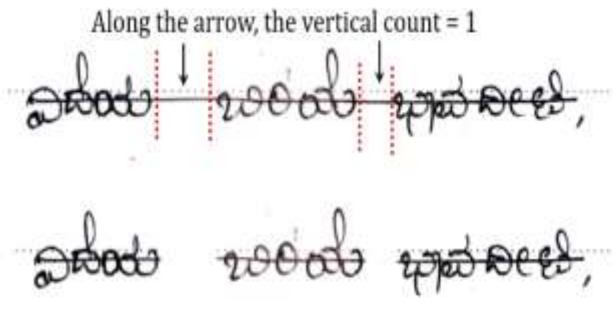


Figure 8. *Two to three horizontal SSs on a word.*

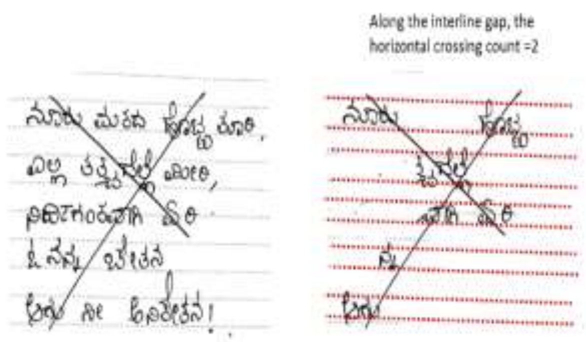
4.10 Handling multi-word and multi-line strike-out

This is one more case where the user can strike out more than one consecutive word in a single stroke. In this condition, AC is created whose B length is literally greater than the height of the word. In this method, a graph-based method which uses directly on such component is used. It is an expensive computation because of many nodes and edges in the graph structure has to be encountered. So, the long component is now split into Multiple shorter and small components. Then they are used in graph-based scheme in each of the possibility.

The splitting can be calculated by counting the components that are vertically crossing. This is shown in the Figure 9. A single line or 2 line passes between 2 words. Here the crossing count of one or 2 line is detected. And hence consecutive single or double vertical crossing count are found. The region can be split vertically as a component. The figure 7 shows a long run off single crossing count. The regions between the dashed verticals lines are a pair. The deletion of lines between the dashed line segment throughout the strike out word is also shown in the same Figure 9.



(a)



(b)

Figure 9. *A single strike throughout multiple words: (a) Along the arrow, the vertical count is 1, (b) Along the interline gap, the horizontal crossing count is 2.*

4.11 Deletion of Strike-out Stroke

As mentioned in the earlier section, the different strike out types are mentioned, they can be recovered for the reader's purpose. If the reader is very much interested in reading all the ideas or the thoughts of the eminent writer. This can be a fair transcription of the manuscript. The graph-based approach is used to detect the skeletal structure of SS. This approach is morphologically directed to obtain the real SS. if does SS cut through the characters stroke, it can be detected by intersecting the text pixels. This can also be used to retain the text pixel fail leave well. And the rest of the SS is deleted. Suppose let us assume this is there is overlapped on the text tangentially then long Porter of the character stroke is difficult to arise. One way to handle this situation each to measure this thickness of stroke.

5. Results and Discussion

Experimental result obtained during the conduction of the experiment and the performance of the method for strikeout text detection and their effect Is mentioned in this section.

1. Normal and strike out text to separation when used the hybrid combination of CNN and SVM:

To proceed with this process, a pre-processing stage for the main task was required and been analysed. The hand return, strike out and writer identification is also performed using graph-based methodology. The evaluation of the performance was carried out using CNN and SVM hybrid classifier. And it was compared with the normal strike out class detection. In the training purpose, 50% of the data was used as generated database and 20% of data was used for training the epochs and finally test it. The CNN was trained using the stochastic gradient descent.

With a learning rate of 10^{-3}. The obtained momentum was 0.9 for CNN. The hyper parameters are also tuned by the training set. The grid section for the γ and were $[2^3, 2^2,..., 2^{-7}]$ and $[2^7, 2^6,..., 2^{-4}]$, respectively the best performance result was obtained for the following cases: $\gamma = 2^{-3}$ and $\gamma = 2^{-6}$. The validation for this purpose was used as 5-fold cross validation. The quality parameters that are used to Measure the accuracy are recall, F measure and precision.

Before defining the nature of precision, the recall and F measure, let us understand what the definition for true positive, false negative, false positive and true negative values is. True positive is number of genuine strikes out words directed by our system. False negative indicates the number of strike out words that are wrongly recognised as strike out words by the system. False positive are the normal words that are detected as strike out and they are strike out. True negative are the words that are strike out, but they are represented as normal. The mathematical representation of precision, F-measure and recall are:

$$Precision\ (P) = TP/(TP + FP) \qquad (4)$$

$$Recall\ (R) = TP/(TP + FN) \qquad (5)$$

$$F\text{-}Measure\ (FM) = (2 \times P \times R)/(P + R) \qquad (6)$$

The quantitative performance of the experimental results is shown in the Table 1. The accuracy obtained for strike out text was 98.69%. Whereas for Kannada text it was 98.41%. The F measure for the English text in standard database was 98.85% and for Kannada language it was 98.85%.

2. Comparison of the proposed methodology with the existing state of art techniques:

The 3 parameters that are considered from the previous paper are F-measure, precision and recall. The obtained results are tabulated in Table 2. The proposed work aims to elimination of a strike out a text in the given data.

Table 1: *A comparison of strike out a text detection.*

Script	Database	Untrained database-Kannada
Kannada	Precision %	98.45
	Recall %	98.93
	F-measured %	98.69

Table 2. *Comparison of strike out text detection by proposed methodology and existing methodology.*

Script	Database	Untrained database-Kannada
Kannada	Precision %	90.94
	Recall %	92.18
	F-measured %	91.56

6. Conclusion

The significant contribution of the present research carried in this article can be summarized as identification of struck out of Kannada text, separation of text, employable of the classifier such as SVM for the classification of the struck-out Kannada text. The key contributions of this research can be summarized as follows:

- Identification of Strikeout Text: The methodology effectively identifies strikeout text in Kannada scripts.
- Text Separation and Classification: The text was separated, and an SVM classifier was employed to distinguish between normal text and strikeout text accurately.

The work was conducted on two distinct datasets:

1. Trained Database: A pre-trained dataset for Kannada text.
2. Untrained Database: Data generated by converting English text to Kannada.

The proposed methodology utilized Python programming with appropriate libraries in the Anaconda environment. Performance metrics such as accuracy, precision, and F1-score were employed to evaluate and compare the effectiveness of the proposed approach with existing techniques.

As shown in Table 1, the existing methods for both the trained and untrained English-to-Kannada converted datasets yielded comparable values for the three metrics. However, the proposed methodology, as detailed in Table 2, achieved superior results for the standard trained Kannada database compared to the untrained database.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1]Namboodiri, A. M., & Jain, A. K. (2004). Online handwritten script recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(1), 124-130. DOI: 10.1109/TPAMI.2004.1261096

[2]Ghosh, D., Dube, T., & Shivaprasad, A. (2010). Script recognition—a review. *IEEE Transactions on pattern analysis and machine intelligence*, *32*(12), 2142-2161. DOI: 10.1109/TPAMI.2010.30

[3]Pal, U., & Chaudhuri, B. B. (2004). Indian script character recognition: a survey. *pattern Recognition*, *37*(9), 1887-1899. DOI: https://doi.org/10.1016/j.patcog.2004.02.003

[4]Kim, I. J., & Kim, J. H. (2003). Statistical character structure modeling and its application to handwritten Chinese character recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(11), 1422-1436. DOI: 10.1109/TPAMI.2003.1240117

[5]Hanmandlu, M., Agrawal, P., & Lall, B. (2009). Segmentation of handwritten Hindi text: A structural approach. *International Journal of Computer Processing of Languages*, *22*(01), 1-20. DOI: https://doi.org/10.1142/S1793840609001993

[6]Voigtlaender, P., Doetsch, P., & Ney, H. (2016, October). Handwriting recognition with large multidimensional long short-term memory recurrent neural networks. In *2016 15th international conference on frontiers in handwriting recognition (ICFHR)* (pp. 228-233). IEEE. DOI: 10.1109/ICFHR.2016.0052

[7]Adak, C., Chaudhuri, B. B., & Blumenstein, M. (2017, May). Impact of struck-out text on writer identification. In *2017 international joint conference on neural networks (IJCNN)* (pp. 1465-1471). IEEE. DOI: 10.1109/IJCNN.2017.7966025

[8]Adak, C., & Chaudhuri, B. B. (2014, September). An approach of strike-through text identification from handwritten documents. In *2014 14th International Conference on Frontiers in Handwriting Recognition* (pp. 643-648). IEEE. DOI: 10.1109/ICFHR.2014.113

[9]Shi, B., Bai, X., & Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, *39*(11), 2298-2304. DOI: 10.1109/TPAMI.2016.2646371

[10]Marti, U. V., & Bunke, H. (2002). The IAM-database: an English sentence database for offline handwriting recognition. *International journal on document analysis and recognition*, *5*, 39-46. DOI: https://doi.org/10.1007/s100320200071

[11]Hegde, S., KK, A., & Shetty, S. (2012, August). Isolated word recognition for Kannada language using support vector machine. In *International conference on information processing* (pp. 262-269). Berlin, Heidelberg: Springer Berlin Heidelberg. DOI: https://doi.org/10.1007/978-3-642-31686-9_31

[12]Ramesh, G., Kumar, S., & Champa, H. N. (2020, June). Recognition of Kannada handwritten words using SVM classifier with convolutional neural network. In *2020 IEEE Region 10 Symposium (TENSYMP)* (pp. 1114-1117). IEEE. DOI: 10.1109/TENSYMP50017.2020.9231003.

[13]Likforman-Sulem, L., & Vinciarelli, A. (2008). Hmm-based offline recognition of handwritten words crossed out with different kinds of strokes.

[14]Brink, A., van der Klauw, H., & Schomaker, L. (2008, January). Automatic removal of crossed-out handwritten text and the effect on writer verification and identification. In *Document Recognition and Retrieval XV* (Vol. 6815, pp. 79-88). SPIE. DOI: https://doi.org/10.1117/12.766466

[15]Chaudhuri, B. B., & Adak, C. (2017). An approach for detecting and cleaning of struck-out handwritten text. *Pattern Recognition*, *61*, 282-294. DOI: https://doi.org/10.1016/j.patcog.2016.07.032

[16]Manmatha, R., & Srimal, N. (1999, September). Scale space technique for word segmentation in handwritten documents. In *International conference on scale-space theories in computer vision* (pp. 22-33). Berlin, Heidelberg: Springer Berlin Heidelberg. DOI: https://doi.org/10.1007/3-540-48236-9_3

[17]Elizondo-Leal, J. C., & Ramirez-Torres, G. (2010, September). An exact euclidean distance transform for universal path planning. In *2010 IEEE Electronics, Robotics and Automotive Mechanics Conference* (pp. 62-67). IEEE. DOI: 10.1109/CERMA.2010.93

[18]Jaramillo, J. C. A., Murillo-Fuentes, J. J., & Olmos, P. M. (2018, August). Boosting handwriting text recognition in small databases with transfer learning. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)* (pp. 429-434). IEEE. DOI: 10.1109/ICFHR-2018.2018.00081

[19]Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006, June). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning* (pp. 369-376). DOI: https://doi.org/10.1145/1143844.1143891

[20]Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Proceedings of the Soviet physics doklady*.

[21]Plamondon, R., & Lorette, G. (1989). Automatic signature verification and writer identification—the state of the art. *Pattern recognition*, *22*(2), 107-131. DOI: https://doi.org/10.1016/0031-3203(89)90059-9

[22]Schomaker, L. (2007, September). Advances in writer identification and verification. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)* (Vol. 2, pp. 1268-1273). IEEE. DOI: 10.1109/ICDAR.2007.4377119

[23]He, Z., You, X., & Tang, Y. Y. (2008). Writer identification of Chinese handwriting documents using hidden Markov tree model. *Pattern Recognition*, *41*(4), 1295-1307. DOI: https://doi.org/10.1016/j.patcog.2007.08.017

[24]Adak, C., & Chaudhuri, B. B. (2015, August). Writer identification from offline isolated Bangla characters and numerals. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)* (pp. 486-490). IEEE. DOI: 10.1109/ICDAR.2015.7333809

[25]Daniels, Z. A., & Baird, H. S. (2013, August). Discriminating features for writer identification. In *2013 12th International Conference on Document Analysis and Recognition* (pp. 1385-1389). IEEE. DOI: 10.1109/ICDAR.2013.280

[26]Tomai, C. I., Zhang, B., & Srihari, S. N. (2004, August). Discriminatory power of handwritten words for writer recognition. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* (Vol. 2, pp. 638-641). IEEE. DOI: 10.1109/ICPR.2004.1334329

[27]Zhang, B., Srihari, S. N., & Lee, S. (2003, August). Individuality of handwritten characters. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* (Vol. 3, pp. 1086-1086). IEEE Computer Society. DOI: 10.1109/ICDAR.2003.1227824

[28]Dintakurthy, Y., Innmuri, R. K. ., Vanteru , A. ., & Thotakuri, A. (2025). Emerging Applications of Artificial Intelligence in Edge Computing: A Comprehensive Review. *Journal of Modern Technology*, *1*(2), 175-185.

[29]Schomaker, L., & Bulacu, M. (2004). Automatic writer identification using connected-component contours and edge-based features of uppercase western script. *IEEE transactions on pattern analysis and machine intelligence*, *26*(6), 787-798. DOI: 10.1109/TPAMI.2004.18

[30]Jain, R., & Doermann, D. (2013, August). Writer identification using an alphabet of contour gradient descriptors. In *2013 12th International Conference on Document Analysis and Recognition* (pp. 550-554). IEEE. DOI: 10.1109/ICDAR.2013.115

[31]P. Rathika, S. Yamunadevi, P. Ponni, V. Parthipan, & P. Anju. (2024). Developing an AI-Powered Interactive Virtual Tutor for Enhanced Learning Experiences. *International Journal of Computational and Experimental Science and Engineering,* 10(4). https://doi.org/10.22399/ijcesen.782

[32]K. Tamilselvan, , M. N. S., A. Saranya, D. Abdul Jaleel, Er. Tatiraju V. Rajani Kanth, & S.D. Govardhan. (2025). Optimizing data processing in big data systems using hybrid machine learning techniques. *International Journal of Computational and Experimental Science and Engineering,* 11(1). https://doi.org/10.22399/ijcesen.936

[33]Wang, S., & Koning, S. bin I. (2025). Social and Cognitive Predictors of Collaborative Learning in Music Ensembles. *International Journal of Computational and Experimental Science and Engineering,* 11(1). https://doi.org/10.22399/ijcesen.806

[34]S. Esakkiammal, & K. Kasturi. (2024). Advancing Educational Outcomes with Artificial Intelligence: Challenges, Opportunities, And Future Directions. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.799

[35]Bertolini, D., Oliveira, L. S., Justino, E., & Sabourin, R. (2013). Texture-based descriptors for writer identification and verification. *Expert Systems with Applications*, *40*(6), 2069-2080. DOI: https://doi.org/10.1016/j.eswa.2012.10.016

[36]Ajith, S. (2024). *A comprehensive review of online and offline handwritten kannada character recognition* (pp. 77–83). https://doi.org/10.58532/v3bkai11p4ch1

[37]Tiptur Parashivamurthy, S. P., & Rajashekararadhya, S. V. (2024). Intelligent Character Recognition Framework for Kannada Scripts via Long Short Term Memory with Thresholding-based Segmentation. *Advances in Artificial Intelligence and Machine Learning*, *04*(03), 2517–2534. https://doi.org/10.54364/aaiml.2024.43147