Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)

An Intelligent Approach for Recognition and Deletion of Strike-Out Text in Kannada Handwritten Scripts

Bhargav H. K.[1]

[1] *Director, IIIC and Assoc.Professor, Department of Computer Science and Engineering,*

Shridevi Institute of Engineering and Technology,Tumakuru-572 106

Abstract: Handwritten document image processing is more demanding and specific area in machine and pattern recognition field. This research work presents a new approach to identify strike –outs in Kannada handwritten document images. To address these issues, paper proposes a hybrid model combining the feature based classifier and graph-based techniques. The dataset is categorized into two classes: striked-out and non striked-out, and a Support Vector Machine (SVM) classifier is used for feature extraction in pattern classification. In the graph-based approach, we employ the shortest path algorithm to analyse strokes and also addressed the complexities of zigzag or wavy style strike-outs.

Keywords: Strike-Out Text processing , Handwritten OCR , Pattern Recognition

1. Introduction

Handwritten documents hold a unique cultural and historical significance in various linguistic communities, including Kannada, a major South Indian language. The preservation of such a ancient scripts which are rich in heritage and knowledge are required to retain for future generation. In this view, research to digitize the handwritten scripts with a advanced computer vision approaches are more challenging and demanding. Especially, handwritten Kannada document images often present challenges, particularly when they include strike-out text, which complicates their preservation, digitization, and readability. It has been found in several handwritten scripts a strike-out text may include character, word ,line and phrase etc are refers to the intentional act. It is a common feature in handwritten texts, used for various reasons such as corrections, revisions, or annotations. Recognizing and deciphering strike-out text is crucial to accurate digitization for transcription, preservation, and understanding the content of Kannada handwritten document(KHD) images. The importance of strike-out detection and recognition in handwritten Kannada documents cannot be overstated.

One of the key challenges in strike-out text recognition in KHD lies in the diverse forms and styles of handwritten Kannada script. The script exhibits an array of individualistic variations, making it difficult to develop a one-size-fits-all solution for strike-out detection. In Addition, strikes-out text can vary in terms of the degree of obliteration, the choice of symbols or marks used, and the context within which they appear. On understanding and learning of these complexities, there is a need to design and develop effective methodologies for detecting and recognizing strike-out text in Kannada handwritten documents. These methodologies intended to enhance the readability and interpretability of KHD and also facilitate the efficient digitization and preservation of this valuable cultural and linguistic heritage. For the understanding , Figure 1 illustrates various styles of strikeout strokes, including single, multiple, slanted, crossed, zigzag, and wavy strokes. The commonly observed strikeout stroke styles are as below:

1. Single Stroke: A single horizontal line is drawn over the word, representing the strikeout stroke.
2. Multiple Strokes: Multiple horizontal lines are drawn over the word to indicate strikeout strokes.

3. Slanted Stroke: A slanted line is used as the strikeout stroke. It can be a single slant line or multiple slant lines drawn on the word. The angle of the slant line typically ranges from 60 degrees to 10 degrees. The slant line may extend above and below the word.
4. Crossed Stroke: A crossed stroke consists of intersecting lines drawn over the word, forming an "X" shape.
5. Zigzag Stroke: Instead of a straight strikeout line, a zigzag line is used to assist in striking out the word. The zigzag line may also have a wavy nature, adding variation to the strikeout stroke. This technique helps partially hide the written word.

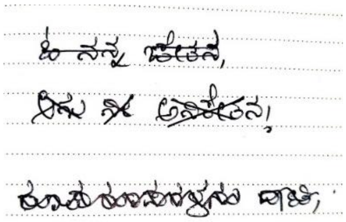


Figure 1: Various forms of strike-out strokes

The presence of these different strike-out stroke structures in handwritten Kannada literature poses a challenge for accurate strikeout detection.

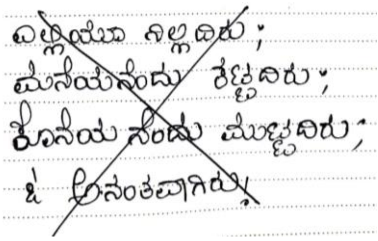


Figure 2: The SSs of complete paragraph and the page

In Figure 2 , a complete paragraph of text is shown with clear Strike-out Stroke(SS) of complete paragraph and the page. The analysis of handwriting reveals that consecutive words, entire paragraphs, or even entire pages can be struck out. To address this challenge, the paper proposes a comprehensive approach to categorize the handwriting script from Kannada alphabets, as well as identify struck out characters and words in the written text.

2. Literature Review

The recognition of strike-out text in handwritten documents holds significant importance in the field of digital document analysis, enabling enhanced transcription and interpretation of handwritten content. This section presents a comprehensive review of existing literature on the recognition of strike-out text in Kannada handwritten documents, encompassing the characteristics of Kannada letters, various types of strike-outs, and the challenges associated with their detection.

The literature reveals a variety of strike-out techniques used in Kannada literature, aimed at crossing out or partially obscuring written content. These include single strikes, multiple strikes, slanted strikes, crossed strikes, zigzag strikes, and wavy strikes. Each type of strike-out stroke involves specific patterns and characteristics that require sophisticated recognition techniques for accurate detection.

Detecting strike-out text in Kannada handwritten documents presents several challenges due to the intricate nature of the script and the diversity of strike-out patterns. One significant challenge lies in differentiating strike-out strokes from legitimate strokes that form part of the script itself. Additionally, the presence of multiple types of strike-outs within a single document further complicates the recognition process. The varying angles, lengths, and curvatures of strike-out strokes demand robust algorithms capable of discerning subtle differences.

Moreover, the potential interference of other handwriting elements, such as underlining or marginal annotations, adds another layer of complexity. The detection of strike-out text must also account for variations in writing styles, ink densities, and paper textures, which impact stroke appearance and contrast. The challenge of accurately segmenting strike-out strokes from surrounding content is also noteworthy, particularly when dealing with overlapping or closely spaced characters.

In a related work [14], a widely adopted method for handwritten script recognition is presented. The author categorizes handwriting recognition systems into two overarching types: those based on visual appearance and those based on structural characteristics. Additionally, [15] offers a concise overview of the diverse scripts and categories employed in their methodology. The author proposes a model for comprehending and identifying text in videos and online content. It is worth highlighting that the domain of handwritten document analysis is continually evolving, necessitating further advancements to address its unique attributes, as emphasized by the author. [15] commences by delving into existing optical character recognition (OCR) systems and their correlation with symbol identities and individual characters. The functionality of the script recognizer is elucidated with a focus on Urdu and English languages. Furthermore, the paper explores various languages utilizing the Brahmi script. The study also touches upon the prevalent script recognition methodologies employed within machine learning environments. The Spitz method, employed in this study, is elucidated through a clear block diagram, highlighting its capability to recognize multiple languages. The system achieved successful recognition of languages such as English, French, German, as well as languages with distinct shapes and varying optical densities like Chinese, Japanese, and Korean. Language recognition was accomplished based on pen position, with samples collected at a rate of 132 samples per second. The algorithm also accounted for stroke patterns and similarities. To enhance image quality, a Gaussian filter was applied. The X-axis projection of the text was utilized for analyzing word segmentation. However, it's important to note that this segmentation approach might encounter challenges when dealing with different regions of handwritten text containing diverse scripts. The Daubechies filter to establish a wavelet domain for multi-resolution analysis of handwritten numeral images. Their multi-stage recognition scheme involved cascaded Multilayer Perceptrons (MLPs). The first stage employed three MLPs for initial image classification, with only successful classifications advancing to the next stage. The second stage utilized another MLP for further classification. Accuracy evaluation using a dataset of 20 classes of handwritten Devanagari data yielded an accuracy of 70.85%. For instances of English mixed with Devanagari script, classification accuracy reached 65.02%. [17] focuses on statistical attributes like pixel density and zero crossing vertex points to differentiate Devanagari characters. Special attention is given to vertical bars and upper modifier boxes. Leverage density, moment features, and a connectionist architecture with multiple classifiers to achieve an accuracy of 89.68% in identifying Devanagari numerals. The paper also explores the use of hidden Markov models for offline word recognition, specifically addressing challenges posed by superimposed strokes without additional identification or cleaning.

3. Proposed Methodology

The initial stage of our approach entails identifying individual lines within the handwritten Kannada document. This involves segmenting the document into discrete lines, facilitating subsequent analysis which follows line identification within regions of strike-out text. Our focus is on pinpointing areas where text has been striked-out, and subsequently, to isolate the connected components within these regions for further process. To enhance document clarity, need to discard diminutive connected components that may lack substantial textual content. Once minor components are filtered out, then isolate and separate prominent connected components. This separation aids to distinguish between the primary body of text and possible annotations or comments.

Here it employed a training of images using Support Vector Machine , is capable to distinguish between Strike-out and non strike-out texts in Kannada handwritten document images based on distinct features . Further a new method is included to discard heavily inked strike-out areas, to ensures only legible and discernible strike-out text is considered for further analysis. Within the realm of strike-out text, our focus shifts to identifying individual strike-out strokes. This entails detecting distinct lines or strokes that constitute the strike-out annotations. Once strike-out strokes are identified then proceed to the phase of strike-out deletion. In this stage, we apply inpainting techniques to intelligently substitute strike-out annotations with appropriate content, effectively "filling in" the struck-out regions with coherent text that aligns with the surrounding context.

The workflow of this paper is represented in Figure 4.

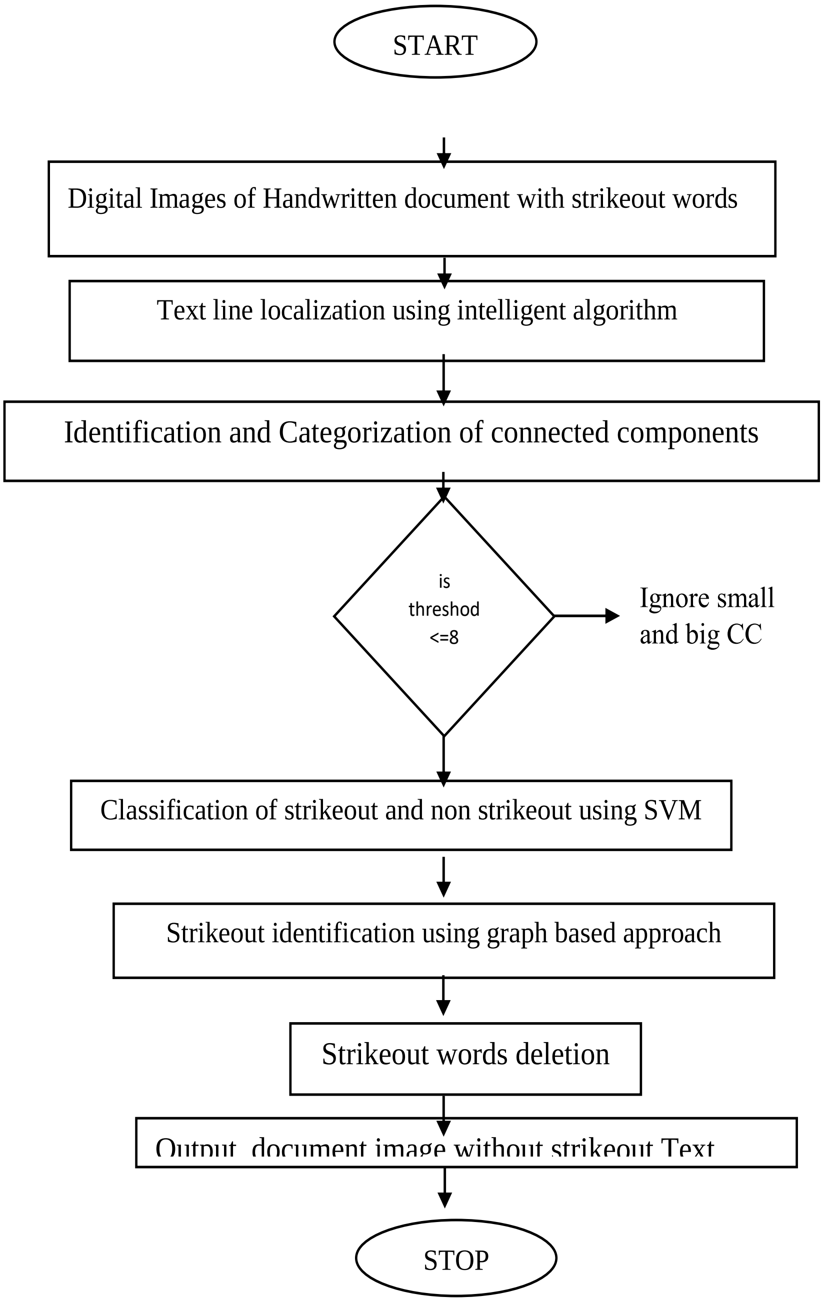


Figure 4: Workflow of Proposed Approach

The process is initiated by employing a pre-existing algorithm to identify individual lines of text within the handwritten Kannada document. This algorithm is adept at handling situations where strike-out text are present. Subsequently, we isolate and analyze the Connected Components (CCs) within each identified line. Minuscule elements like dots, dashes, commas, colons, and noise are excluded from subsequent processing stages.

Acknowledging that struck-out content usually constitutes a small fraction of the total word count, we leverage the average height (H_{av}) of Bounding Boxes (BBs) encompassing these components. H_{av} serves as a dependable gauge of the typical word height. Additionally, we calculate the standard deviation (H_{sd}) of BB heights for these components.

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)

Continuing the procedure, the length of each BB is computed. CCs meeting the criterion of BB lengths within the range of H_{av} to αH_{av} and heights within $H_{av} \pm H_{sd}$ are identified as instances of type (a), as defined previously. These instances then undergo scrutiny through the Support Vector Machine (SVM) for analysis.

In our approach, we utilize a multiplier denoted as α (where $\alpha > 1$), and for our specific case, α is set to 8. This choice of α has been determined through empirical observations, particularly in the context of Kannada scripts. These scripts typically consist of words containing fewer than 9 characters or ortho-syllables.

When applying our method, we have found that using $\alpha = 8$ yields more favorable results. This is primarily due to the fact that most words in Kannada scripts tend to be shorter than 9 characters/ortho-syllables. However, it's important to note that if a continuous character sequence (CC) surpasses this particular length threshold, it falls into a distinct category referred to as case (b).

In a broader sense, both case (b) and case (c) are characterized by larger CCs. These larger CCs are subsequently transformed into medium-sized CCs through the process of selectively removing strokes. Once this transformation is applied, the resulting medium-sized CCs are then input into an SVM classifier.

The classifier we utilize is designed to differentiate between two distinct classes: normal components, which are not subjected to strike-outs, and strike-out components that have undergone alteration. To extract the features necessary for the SVM classifier, each continuous character sequence (CC) undergoes a process known as skeletonization. This procedure serves multiple purposes. First and foremost, the primary purpose of this process is to facilitate the identification of components within the character clusters (CCs) that have been categorized by the Support Vector Machine (SVM) as belonging to a class and have undergone multiple strike-outs, rendering them impossible to restore to their original state. This identification is accomplished by quantifying the number of thinning iterations necessary to produce the skeleton of the component. In cases where a word has experienced extensive marking, a greater number of iterations is needed compared to an unmarked word. Additionally, in such situations, the density of black pixels within the bounding box (BB) of the component surpasses that of a normal word. These distinct characteristics enable the detection of heavily marked words that are no longer suitable for further processing. Moreover, this procedure effectively eliminates extraneous black ink-blobs that may be present in the document. The second objective of the skeletonization process is to create a graph-like representation of the component. This representation forms the foundation for identifying strike-outs. To detect instances of strike-outs, the component's skeleton is transformed into an undirected graph. Our approach entails searching for nearly horizontal shortest paths (SPs) that connect a node on the left side to another node on the right side of the graph. The presence of such a path signifies that a word has been struck out. Recognizing that the computation of SPs across various nodes in all components within a document can be computationally demanding, we initially employ a feature-based SVM classifier. This classifier expedites the identification of components that have been struck out, streamlining the processing and analysis.

3.1 Identification of Strike-out Stroke

We employ a graph-based approach to efficiently detect strike-through segments (SS) within the connected component of a struck-out word. When dealing with a text component image (I), we create its skeleton (Isk) using a thinning algorithm. To eliminate any extraneous branching in the skeleton, we follow these steps:

1. We identify skeletal pixels with only one neighbouring pixel, which we term 'end' or 'terminal' pixels.
2. We also identify skeletal pixels with three or more neighbouring pixels, forming three or more branches, which we refer to as 'junction' pixels.
3. If a direct path exists from a terminal pixel to a junction pixel, and the length of this path is shorter than half of the average stroke width (ST) of the original connected component, we proceed to remove the pixels along this path, including the terminal pixel.

While the thinning algorithm is used for generating the skeleton. Following the pruning of spurious branches as outlined, the resulting shape of the skeleton remains consistent across these effective algorithms. This uniformity arises because handwritten text strokes typically possess a much smaller width in comparison to their length, resulting in highly elongated strokes. Consequently, a well-designed approach for generating the skeleton yields a clean representation that is largely devoid of erroneous branching.

An ordered pair G = (V,E), where V signifies the set of nodes, and E represents the set of edges where skeleton represents.

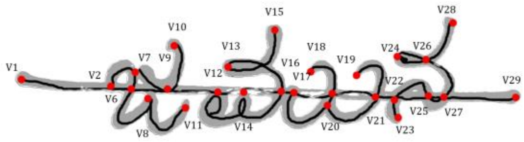


Figure 3: divides THREE regions

The Figure 3 shows the skeleton image which is divided into 3 regions. The left region, right region and the mid region. The connection between V_1 and V_{28} show the SS.

3.2 Nodes identification

Nodes are in the graph G is shown by the terminal pixels and junction pixels present in the skeleton Isk where nodes are positioned closely to each other (with a distance less than Tk pixels), for a single node that is left-to-right movement while path estimation. The value of Tk is determined through the ceiling function on 1/2 of the avg stroke width (ST) of the image component I. Tk equals to ⌈ST / 2⌉. Tk adapts to variations in the stroke-image resolution, which in turn influence the average stroke width.

3.3 Identification of Edges (E)

Edges (e_{ij}) are established between pairs of node pixels (vi and vj, where vi and vj belong to V) within the graph. These edges arise when the nodes are connected solely by non-node object pixels. It's feasible for multiple edges to exist between the same pair of nodes, and in such cases, these edges are labelled as e_{ij1}, e_{ij2}, and so forth. Additionally, a node might possess a self-loop, which is represented as e_{iL}. In order to establish the weight assigned to an edge (wij), we consider both the number of diagonal moves (Nd) and the number of horizontal or vertical moves (Nhv) needed to move from node vi to node vj while exclusively traversing through object pixels. The weight of the edge (wij) is determined based on these traversal counts.

$$\omega_{ij} = \omega_d N_d + w_{hv} N_{hv} \quad \text{....eq(2)}$$

3.4 Shortest Path

Previous stage, it is considered to have a reasonable straight SS in the document. 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.

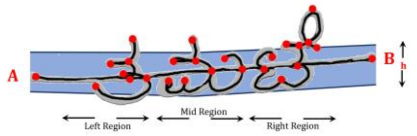
3.5 Speed up Approach.

FOUR steps to analyse the strike outs in speed up approach.

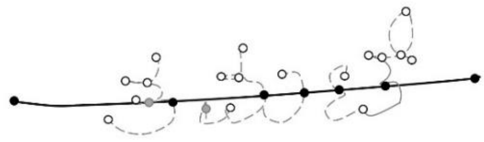
i. To divide the boundary box into 3 equal components. The 3 equal components are named as a right region, left region and the mid region. In the Figure 4, it is clear that normally the strike out is from left to right. And hence we can avoid the middle region nodes as those are not in the shortest path.

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.

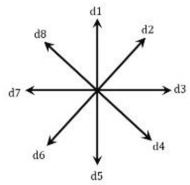
Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)



(a)



(b)



(c)

Figure 5: (a) a to b Path, with the band considered with Height (b) Generation of a shortest path using band. (c) 8- neighbour 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 Fig 5(a), the skeleton structure of the node-edge graph is displayed. The black nodes and the grey nodes represent the pathway from point A to point B. The white nodes are situated outside this pathway or band. It's important to note that only the black nodes play a role in the eventual generation of the shortest path.

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.

3.6 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 objects. 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.

3.7 Recognising a wavy stroke or a zigzag stroke.

One of the difficult and challenging situation in the strikes stroke recognition is when it encounters Azad stroke. The zigzag strokes or the wavy strokes are shown in Figure 6.

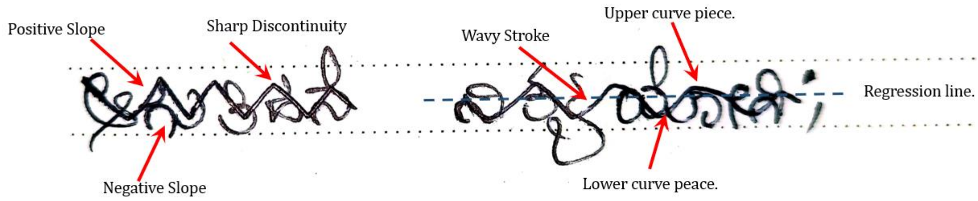


Figure 6: Processing of wavy strike out and zigzag strike out.

The strikeout stroke is indicated in red color. In this particular scenario, we cannot apply the shortest path scheme since the path is neither straight nor short. However, it's essential to recognize that there are distinct components or segments within 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.

To speed up the approach, the number of parts are 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.

3.8 Recognizing non-horizontal strike-outs

Another example of striking out the words is a slant strike, which is shown in Figure 7. 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.

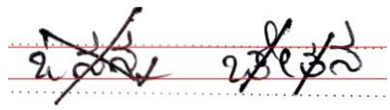


Figure 7: Direction of near vertical and strike outs.

In Figure 7, the slope of summer strike outs are 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]

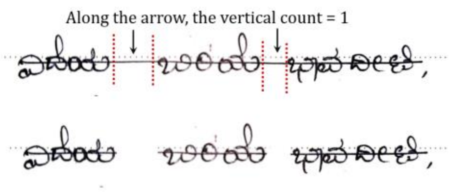
3.9 multiple and script-specific strike strokes

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: 2 to 3 horizontal SSs on a word.

3.10 multi-words and multi-lines strike-outs

This situation presents an opportunity for users to eliminate multiple consecutive words with a single gesture. Under this circumstance, an AC is formed, with a "B" length that significantly surpasses the word's height. To address this challenge, a graph-based approach is employed, focusing directly on these components. This method involves computationally intensive processes due to the substantial number of nodes and edges within the graph structure. Consequently, the elongated component is divided into several smaller, more manageable segments, each of which is then integrated into the graph-based framework, exploring all potential configurations.



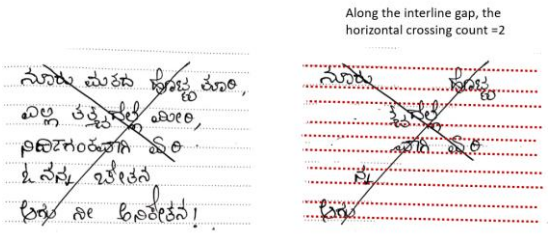


Figure 9: A single strike-out throughout multiple words.

The splitting can be calculated by counting the components that are vertically crossing. This is shown in the Figure 9. A single line or TWO line passes between TWO words. Here the crossing count of ONE or TWO 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 is a pair. The deletion of lines between the dashed line segment throughout the strike out word is also shown in the same Figure 9.

3.11 Deletion of Strike-out Stroke

The paper discusses the use of a graph-based approach to detect the skeletal structure of Strike-out Strokes (SS) in a manuscript. By intersecting the text pixels, the approach can identify where the SS cuts through the stroke of characters, allowing for the retention of text pixels and removal of the rest of the SS. If there is overlap with the text tangentially, the thickness of the stroke is measured and uniformity is achieved using an inpainting approach, which fills the gaps in the strikeout structure. The inpainting technique involves selecting a mask and searching for the target region.

4. Results and Discussions

The results obtained from the experimental phase, which was conducted to evaluate the effectiveness of the method for detecting strikeout text, are discussed in this section. We also delve into the impact of these results on the performance of the method.

4.1 Preparation of Dataset

The database we utilized for this study consisted of writing samples from around 100 Kannada medium school students, who were roughly 10 years old, with an age range of about ± 3 years. These students were all from Karnataka and were proficient in the Kannada language, having had at least 4 years of writing experience. To gather the samples, we provided the students with information about the study and asked them to write a passage from the famous Kannada poem "Vishvamanava Sandesha" by the well-known poet Sri Kuvempu. The writing was done by hand and included strike-out text. We employed a suitable scanner to convert the handwritten samples into digital format. For further analysis, we processed the collected data using the Python programming language and relevant software libraries.

4.2. Detection of Strike-outs Connected Components

The frequency of strike-out strokes (SS) in the database was examined to gain insights into the various strike-out types present in the writing samples. The table below presents the distribution of these strike-out types among the collected data from the Kannada medium school students. The strike-out types include single, multiple, slanted, crossed, zigzag, wavy, and other forms. Each type reflects the distinct ways in which students corrected or modified their written content. Table 1shows the Frequency of strike-out stroke (SS) types in the database.

Table 1: Frequency of strike-out strokes in the database

Strike-Out Type	Frequency
Single	215
Multiple	120
Slanted	115
Crossed	80
Zigzag	47
Wavy	42
Others	12
Total	631

Table 2: Performance of strike-out word detection

Word length in character	Precision %	Recall %	F-Measure
1	71.00	82.00	76.10
2	74.43	76.57	75.48
3	82.12	83.37	82.74
4	89.00	76.00	81.98
5	88.45	89.76	89.10
More than 5	95.00	88.00	91.00
Overall	83.33	82.61	82.73

The overall precision of 83.33 indicates the accuracy of correctly identifying strike-out words among all detected instances. The overall recall value of 82.61 demonstrates the method's ability to capture the majority of actual strike-out words present in the documents. The calculated overall F-measure of 82.73 presents a balanced assessment, considering both precision and recall, and provides a comprehensive overview of the method's effectiveness in accurately detecting strike-out text in the context of Kannada handwritten documents.

6. Conclusion.

The overall work focuses on identification and deletion of strike-out text in Kannada handwritten document. Support Vector Machine is used for classification of strike out and non strikeout text in handwritten Kannada document resulted in significant results followed by inpainting approach to delete strikeout portion . The study examines both trained and untrained databases using parameters like precision, recall and F1 score. The proposed methodology gives better results for the given database further modified version of algorithms and hybrid models may used to enhance the accuracy in the results.

References.

[1] A.M.Namboodiri and A. K. Jain, “Online script recognition”, IEEE Transactions on Patten Analysis and Intelligence, Vol.26, No.1, pp.124-130, January 2004.

[2] U. Pal and B. B. Chaudhuri, "Indian Script character recognition: A survey", Pattern Recognition, Vol. 37, pp. 1887- 1899, September 2004.

[3] In-Jung Kim and Jin Hyung Kim, "Statistical character structure modeling and its application to handwritten Chinese recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No.11, pp.1422-1436, October 2003.
[4] M. Hanmandlu, Pooja Agrawal and Brejesh Lall, "Segmentation of Handwritten Hindi Text: A structural Approach", International Journal of Computer Processing of Languages, Vol. 22, No. 01, pp.167-173, March 2009.
[5] J. Puigcerver, "Are multidimensional recurrent layers really necessary for handwritten text recognition?" in 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1. IEEE, 2017, pp. 67–72.
[6] P. Voigtlaender, P. Doetsch, and H. Ney, "Handwriting recognition with large multidimensional long short-term memory recurrent neural networks," in 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). IEEE, 2016, pp. 228–233.
[7] C. Adak, B. B. Chaudhuri, and M. Blumenstein, "Impact of struck-out text on writer identification," in 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2017, pp. 1465–1471.
[8] C. Adak and B. B. Chaudhuri, "An approach of strike-through text identification from handwritten documents," in 2014 14th International Conference on Frontiers in Handwriting Recognition. IEEE, 2014, pp. 643–648.
[9] B. Shi, X. Bai, and C. Yao, "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, vol. 39, no. 11, pp. 2298–2304, 2016.
[10] U.-V. Marti and H. Bunke, "The iam-database: an english sentence database for offline handwriting recognition," International Journal on Document Analysis and Recognition, vol. 5, no. 1, pp. 39–46, 2002.
[11] L. Likforman-Sulem and A. Vinciarelli, "Hmm-based offline recognition of handwritten words crossed out with different kinds of strokes," 2008.
[12] Brink, H. van der Klauw, and L. Schomaker, "Automatic removal of crossed-out handwritten text and the effect on writer verification and identification," in Document Recognition and Retrieval XV, vol. 6815. International Society for Optics and Photonics, 2008, p. 68150A.
[13] B. Chaudhuri and C. Adak, "An approach for detecting and cleaning of struck-out handwritten text," Pattern Recognition, vol. 61, pp. 282– 294, 2017.
[14] R. Manmatha and N. Srimal, "Scale space technique for word segmentation in handwritten documents," in International conference on scale-space theories in computer vision. Springer, 1999, pp. 22–33.
[15] J. C. Elizondo-Leal and G. Ramirez-Torres, "An exact euclidean distance transform for universal path planning," in 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference. IEEE, 2010, pp. 62–67.
[16] J. C. Aradillas, J. J. Murillo-Fuentes, and P. M. Olmos, "Boosting handwriting text recognition in small databases with transfer learning," arXiv preprint arXiv:1804.01527, 2018.
[17] Graves, S. Ferna´ndez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 369–376.
[18] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in Soviet physics doklady, vol. 10, no. 8, 1966, pp. 707–710.
[19] R. Plamondon, G. Lorette, "Automatic Signature Verification and Writer Identification - The State of the Art", Pattern Recognition, vol.22, no.2, pp.107-131, 1989.
[20] L. Schomaker, "Advances in Writer Identification and Verification", Proc. Int. Conf. on Document Analysis and Recognition (ICDAR), vol.2, pp.1268-1273, 2007.
[21] Z. He, X. You, Y.Y. Tang, "Writer Identification of Chinese Handwriting Documents using Hidden Markov Tree Model", Pattern Recognition, vol.41, no.4, pp.1295-1307, 2008.