

A Lightweight CNN Framework for Efficient Strike-Out Removal in Handwritten Text Images

Debabrata Khargharia*, Samir Kumar Borgohain

National Institute of Technology Silchar,
Department of Computer Science and Engineering,
India

{debabrata21_rs, samir}@cse.nits.ac.in

Abstract. Due to corrections and revisions, strike-out words are very common in handwritten documents. OCRs that process such words tend to cause recognition errors and low overall accuracy. Current approaches to managing strike-out text are often based on heuristic rules or large pretrained deep networks, which are either not robust or are computationally expensive. The present work proposes a lightweight convolutional neural network, termed StrikeCNN to detect strike-outs at the word level in handwritten documents. The method treats strike-out identification as a binary classification problem and operates directly on binarized word images which allows the model to learn relevant structural patterns without manual feature design. The proposed model achieves an accuracy of 98.8% for strike-out detection on Assamese handwritten words. The experimental results also show significant performance improvements in inference time, training time and model size compared to benchmark architectures. StrikeCNN is simple and efficient, which makes it applicable to the real world in OCR preprocessing pipelines.

Keywords. Handwritten document analysis, strike-out detection, word-level classification, binary classification, OCR preprocessing.

1 Introduction

Significant challenges are incurred during handwritten document analysis due to the inherent variability in handwriting styles, stroke formation, document layout and noise introduced during scanning or image acquisition [11]. As opposed to printed text, handwritten documents often contain irregular structures and corrections made during the writing process which complicates automatic

interpretation. The reliability of document image analysis systems especially optical character recognition (OCR) which typically assumes clean and valid textual input is collectively impacted by these factors [19].

In practical handwritten documents, corrections are very often made by crossing out words or phrases that are irrelevant. While such strike-out text is meaningful to a human reader, it does not contribute useful semantic information for automatic recognition systems. Treating such words in the same manner as valid text can lead to the introduction of recognition errors and reduction in the overall accuracy in the OCR pipeline. Therefore identification and handling of strike-out words is an important preprocessing step for improving the robustness of handwritten document processing systems [20].

1.1 Strike-Out Words

Strike-out words are the handwritten words or phrases that the writer has deliberately rendered invalid by drawing horizontal, diagonal or curved strokes through them. These markings are commonly used to correct spelling mistakes, revise sentence structure or remove unwanted content [20]. Strike-outs can be of different types as shown in Figure 1.

From a document analysis perspective, strike-out words differ structurally from regular handwritten text due to the presence of intersecting strokes and abnormal line patterns superimposed on character shapes [11].



Fig. 1. Different types of Strike-Outs

Most conventional OCR systems implicitly assume that all segmented word images correspond to valid text, which leads to incorrect recognition when strike-out words are present. To address this, earlier approaches have relied on heuristic-based techniques such as straight-line detection, morphological operations or manually designed rules to identify crossed-out text. Although these methods can be effective under controlled conditions, they are highly sensitive to variations in writing style, strike orientation and stroke thickness. As a result, their performance often degrades in unconstrained handwritten documents [11].

More recent studies have explored deep learning-based solutions for handling strike-out text [13, 18]; however many of these approaches employ large pretrained models originally developed for natural image classification. Such architectures introduce unnecessary computational complexity and are not optimized for word-level handwritten document analysis. This highlights the need for a dedicated, lightweight and data-driven approach that can reliably distinguish strike-out and non-strike-out words while maintaining efficiency and scalability.

1.2 Contribution and Organisation

Our work has contributed mainly as follows:

- **Word-Level Formulation of Strike-Out Detection:**
Our work formulates strike-out handling in handwritten documents as a word-level binary classification problem which enables explicit identification and removal of invalid words prior to OCR processing.
- **Lightweight CNN-Based Architecture (StrikeCNN):**
A compact convolutional neural network termed StrikeCNN is proposed for strike-out detection using binarized word images which eliminates the need for handcrafted features or heuristic rules.
- **Cost-Sensitive Learning for Imbalanced Data:**
To address the imbalance between strike-out and non-strike-out words, a loss-weighted binary cross-entropy is used which allows improved detection of minority-class samples while preserving the original data distribution.

- Practical Strike-Out Removal Strategy:
Strike-out words are handled by discarding entire word images, avoiding pixel-level manipulation and simplifying OCR integration.

The paper is organised as follows: Section 2 gives a review of the prior research, which is relevant to this study. Section 3 gives a clear description of the dataset we have used and gives an overview of our proposed methodology. In Section 4, the experimental results and a discussion on results are discussed. Section 5 provides a conclusion and recommendations on possible future research in this area.

2 Related Work

Recent researches have studied different methodologies for handling strike-out text in handwritten documents. Nigam et al. [8] proposed a Variable Cycle Generative Adversarial Network (VCGAN) that removes strike-out strokes from handwritten documents. Their approach works like a digital restoration tool that eliminates strike-out sections while preserving the original handwriting's natural structure and flow. The model successfully restores handwritten text to a cleaner and more readable state. It does this by learning to distinguish between actual valid text and strike-out marks and then reconstructing clean text. This improves readability for both humans and automated recognition systems. The methodology can be useful for digitizing historical documents, processing edited manuscripts or recovering information from strike-out contents.

Nisa [9] made another significant contribution by addressing the same problem which is recognizing handwritten text when the documents consist of both regular writing and strike-out sections mixed together. This work addressed the real world nature of handwritten documents, where often strike-out words, inserted words and overlapped words are present. The research focused on using deep learning models designed specifically for sequence recognition. The study emphasized that handling such chaotic handwriting requires specialized preprocessing

steps and stronger training methods to get decent recognition accuracy.

Zhong et al. [20] proposed a two-stage framework designed to detect and restore strike-out handwritten words in the context of automatic descriptive answer evaluation. Their methodology integrates deep feature extraction through ResNet-based classification with U-Net and Bi-LSTM networks to reconstruct the underlying textual information obscured by strike-out marks. The study demonstrated that the presence of strike-out words considerably degrades recognition accuracy in handwritten text analysis. Consequently, the authors emphasized that effective detection and restoration of such distortions serves as a critical preprocessing step for developing reliable OCR-based evaluation systems, particularly in educational assessment scenarios where non strike-out text must be accurately distinguished and processed.

Urs and Chethan [3] conducted a comprehensive study on identifying and removing strike-out words in handwritten documents. They systematically reviewed detection and cleaning strategies, including connected component analysis, graph-based stroke identification, and machine learning techniques such as SVM-based classification. The authors also explored statistical sequence modeling approaches using hidden Markov models and language modeling for recognizing distorted text. Their findings highlighted the complexity of detecting diverse strike-out patterns across different handwriting styles and emphasized the need for robust preprocessing methods to improve recognition accuracy when distinguishing non strike-out text from intentionally marked content.

Nigam et al. [7] focussed on Indic scripts with limited annotated data for working on the strike-out removal in multilingual handwritten documents. They proposed a transfer learning framework which adapts features from Roman script models to multiple Indic scripts using deep generative models with few-shot and zero-shot learning strategies conducting experiments across 10 Indic scripts and demonstrated that transfer learning reduces training requirements.

Chaudhuri and Adak [2] presented a structured methodology for detecting and removing strike-out text in unconstrained handwritten documents. Their methodology integrates SVM-based pattern classification with graph-based analysis to separate non strike-out text from strike-out words. The strike-out words are analyzed using skeleton-based graph representation where shortest-path analysis is used to identify various strike-out patterns including straight, zigzag and wavy strokes. The approach also incorporates an inpainting method to reconstruct cleaned text after the stroke removal, facilitating improved OCR processing. This multi-stage pipeline demonstrates the effectiveness of combining classification and graph-theoretic methods for handling diverse strike-out patterns in the real-world handwritten documents.

Heil et al. [5] explored strike-out removal using generative image translation techniques based on CycleGAN architectures. They formulated strike-out removal as an image-to-image transformation problem, converting handwritten text with strike-out strokes into clean representations. The framework employs adversarial learning to preserve underlying character structure while suppressing strike-out distortions. By learning mappings between strike-out and valid text samples, the method demonstrated effective restoration performance across different handwriting styles. However, the approach requires extensive training data and introduces higher computational complexity due to its reliance on generative adversarial networks.

Shivakumara et al. [14] proposed a framework for text line segmentation in handwritten documents containing strike-out words. Their method begins by detecting candidate components through stroke-width estimation and morphological processing at the word component level. A DenseNet-based deep learning model then classifies components as either strike-out or non strike-out text. Following classification, spatial and geometric relationships among neighboring components are analyzed to group words into text lines while excluding invalid strike-out content. The framework demonstrated improved segmentation performance across multiple

datasets and writing styles. However, it primarily focuses on enhancing text line segmentation rather than direct word-level strike-out filtering, and the multi-stage processing pipeline introduces increased computational complexity.

Sangamesh et al. [17] proposed a Support Vector Machine (SVM) based methodology for detecting and eliminating strike-out words in handwritten Kannada documents. Their approach combines connected component analysis (CCA) with a SVM classifier to separate normal text from strike-out text, followed by a graph-based shortest path strategy to identify strike-through strokes. To restore the handwritten content, the detected strike-through strokes are removed using an inpainting-based cleaning technique. The method handles multiple strike-out patterns, including slanted, crossed, zigzag and wavy strokes and was evaluated on both trained and untrained Kannada handwriting datasets, demonstrating improved OCR readability after strike-out removal.

A deep learning approach was proposed by Nisa et al. [10] handwritten text recognition in the presence of struck-out text. They studied the impact of struck-out words on handwritten text recognition using a Convolutional Recurrent Neural Network (CRNN). They showed that recognition accuracy degrades when strike-out strokes are present, but retraining the model with struck-out examples enables the network to learn strike patterns and recover recognition performance without affecting overall accuracy.

Bhattacharya et al. [1] proposed a density-based method which focussed on unwanted regions such as strike-out strokes, overwriting and repeated writing in online handwritten text. They proposed density-based features combined with a two-class Hidden Markov Model (HMM) to distinguish relevant text from noisy regions, achieving effective detection without manual intervention.

Shonekov et al. [15] proposed data augmentation techniques to improve handwritten text recognition performance which includes a strike-out simulation technique and a printed text based handwriting generation method.

When these are combined with a ResNet-BiLSTM-CTC architecture, they significantly reduced word and character error

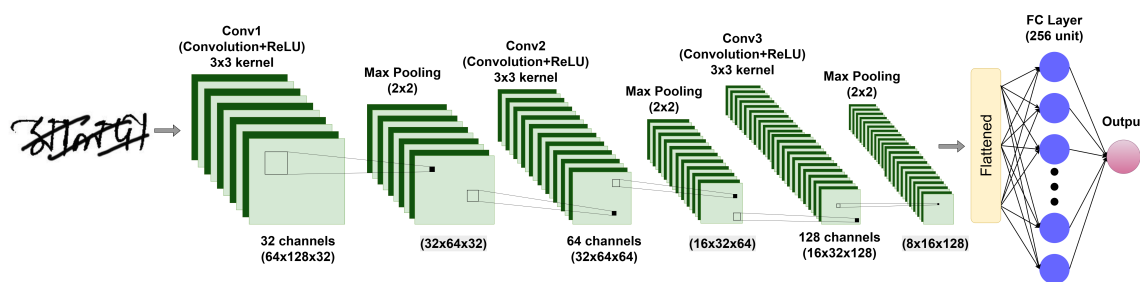


Fig. 2. System Architecture of the proposed StrikeCNN model

rates across different handwritten text datasets which demonstrates their effectiveness as model-agnostic enhancements.

Poddar et al. [12] worked on simultaneous detection of strike-out words and the localization of the strokes by using a GAN-based framework. The method handles a wide variety of strike patterns and was validated on both the IAM dataset and real-world handwritten samples which demonstrates robustness across diverse writing conditions.

Despite recent progress in handling strike-out text, most of the existing approaches focus on recognising and reconstructing the struck-out text or improving recognition after distortion removal. These methods often rely on complex generative or deep pretrained models, which increase computational cost and make real-time deployment difficult. Many approaches attempt to recover strike-out content instead of identifying and eliminating such words during preprocessing phases which highlights the need for a simpler and efficient solution. Our present work addresses this gap by introducing a word-level detection framework that is lightweight and directly distinguishes strike-out and non-strike-out words using binarized inputs enabling fast and reliable filtering without reconstruction or heuristic rules.

3 Proposed Methodology

3.1 Dataset Description & Preprocessing

All the experiments are conducted on Assamese handwritten word images obtained from the OAHTD dataset [6]. This dataset comprises of

different variations in writing style, stroke thickness and strike-out patterns that are commonly observed in real handwritten documents. The word dataset contains 2,505 segmented word images that annotated at the word level into two classes, strike-out and non-strike-out. There are 2,062 non-strike-out words and 443 strike-out words which ultimately leads to a naturally imbalanced class distribution reflecting a natural document correction behavior in which valid words occur far more frequently than strike-out ones.

All the word images are binarized before training to preserve structural stroke information and to reduce the background noise. These images are then resized to 64×128 pixels to maintain a stable input size. This step standardizes the input representation, lowers the computational cost and also allows the network to focus on discriminative stroke and strike-out patterns rather than intensity variations.

3.2 Proposed Model: StrikeCNN

The proposed StrikeCNN is a lightweight convolutional network designed for strike-out detection at the word level. It uses 3 convolutional layers with increasing depth that is followed by ReLU activation and max-pooling to capture structural and stroke-level features. These extracted features are passed through 2 fully connected layers where the final layer produces a single logit for binary classification. This framework enables effective feature learning and also maintains low computational cost making it suitable for OCR preprocessing.

StrikeCNN Architecture Summary

Layer	Type	Kernel / Units	Output Size	Params
1	Conv + ReLU	3×3, 32	32×64×128	320
2	MaxPool	2×2	32×32×64	0
3	Conv + ReLU	3×3, 64	64×32×64	18,496
4	MaxPool	2×2	64×16×32	0
5	Conv + ReLU	3×3, 128	128×16×32	73,856
6	MaxPool	2×2	128×8×16	0
7	Fully Connected	256	256	4,194,560
8	Output Layer	1	1	257

Total Trainable Parameters: 4,287,489

Fig. 3. Overall architectural summary of StrikeCNN

Algorithm 1 Word-Level Strike-Out Detection using StrikeCNN

Require: Segmented word images W , ground-truth labels Y

Ensure: Predicted labels $\hat{Y}$ and filtered non-strike-out words

```

1: Preprocessing:
2: for each  $w_i \in W$  do
3:   Convert to grayscale, normalize to  $[0, 1]$ ,
   resize to  $64 \times 128$ 
4: end for
5: Split data into training, validation and test
   sets (70:20:10)
6: Compute class imbalance ratio from training
   set
7: Initialize StrikeCNN architecture
8: Define weighted Binary Cross-Entropy loss
   and Adam optimizer
9: Training:
10: for each epoch do
11:   for each training mini-batch do
12:     Forward propagation
13:     Compute weighted loss
14:     Backpropagation and parameter update
15:   end for
16: end for
17: Inference and Removal:
18: for each test word image do
19:   Compute sigmoid probability
20:   Assign label using threshold 0.5
21:   if label = non-strike-out then
22:     Retain word image
23:   end if
24: end for
25: Evaluate performance using Accuracy,
   Precision, Recall, F1-score and ROC-AUC

```

Feature Map Transformation: StrikeCNN starts with a 64×128 input binarized word image and then uses a series of convolution and pooling techniques to lower the spatial resolution while raising the number of feature channels. Important word structural and spatial features are learned through this approach. These feature maps are then flattened and fed into the fully connected layers for final classification.

Binary Classification Strategy: Strike-out detection is treated as a binary classification job where each word image is labeled as strike-out or non-strike-out. During inference, a sigmoid activation function is used to produce probability scores. A threshold of 0.5 is applied to determine the final class label. The approach enables direct filtering of invalid words at the word level without the need for stroke removal at pixel-level leaving with only the valid words as output.

Training Strategy and Loss Function: Due to the natural imbalance between strike-out and non-strike-out samples, a weighted binary cross-entropy loss is employed during training. This cost-sensitive learning strategy penalizes misclassification of the minority class more heavily, improving detection robustness without altering the original data distribution.

Model Summary: As shown in Fig. 3, the proposed model is built using three convolutional layers with filter depths of 32, 64 and 128. Each layer employs a 3×3 kernel followed by a ReLU activation function to learn stroke-level and structural features from the binarized word images. Max-pooling layers with a 2×2 window are inserted between the convolutional blocks to gradually reduce spatial resolution while retaining the most informative features. The extracted feature maps are flattened and passed through a fully connected layer with 256 units, which enables compact high-level feature representation prior to classification. The final output layer produces a single logit corresponding to the binary decision between strike-out and non-strike-out words.

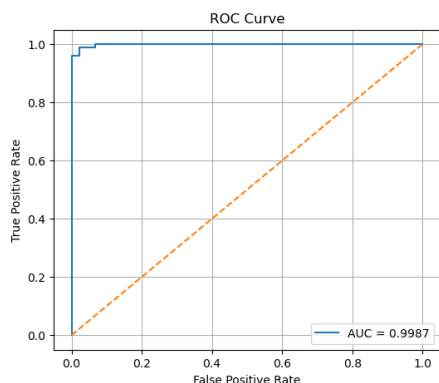


Fig. 4. ROC-AUC curve of StrikeCNN

Overall, the model contains approximately 4.29 million trainable parameters, achieving a balance between representational capacity and computational efficiency, making it suitable for word-level OCR preprocessing tasks. Algorithm 1 summarizes the overall training and inference procedure.

4 Experimental Results and Discussion

In this section, the experimental results achieved with the help of the proposed method of strike-out word classification are provided and the performance of this method is compared to the existing methods. The analysis is done based on the effectiveness of the classification as well as the effectiveness of computing in the realistic scenario of handwritten documents. The balanced F1-score highlights the model's suitability for preprocessing tasks where both missed detections and incorrect removals can negatively impact downstream OCR performance.

4.1 Classification Performance

The proposed model shows strong classification performance in distinguishing the strike-out words from the non strike-out words across the test set. The quantitative classification results obtained on the test set are summarized in Table 1.

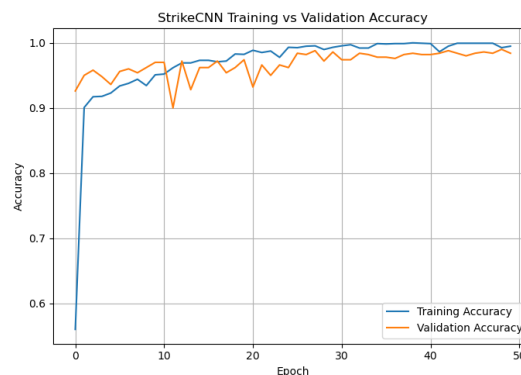


Fig. 5. Training vs Validation Accuracy

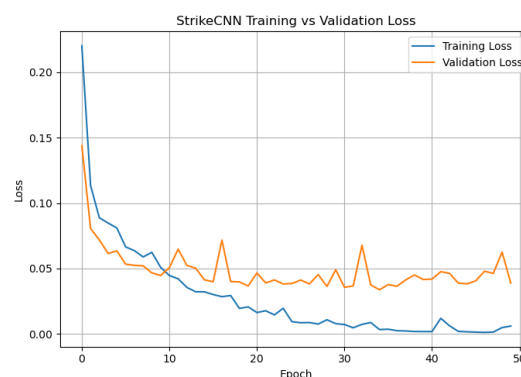


Fig. 6. Training vs Validation Loss

The proposed model achieves an overall accuracy of 98.8% with a precision of 99.29%, recall of 96.59% and a F1-score of 97.88%. These results indicate reliable discrimination between strike-out and non-strike-out word images despite significant handwriting variability.

The ROC-AUC value of 0.9987 (Fig. 4) indicates excellent class separability and confirms the robustness of the proposed model under imbalanced data conditions. Although the dataset is imbalanced, accuracy is reported alongside precision, recall and F1-score to provide a complete evaluation, as false positives and false negatives have distinct implications in OCR preprocessing pipelines.

Words BEFORE Strike-out Removal

Fig. 7. Words before the strike-out removal

Words AFTER Strike-out Removal

Fig. 8. Words after the strike-out removal

4.2 Training and Validation Behaviour

The learning behavior of the proposed StrikeCNN model is illustrated in Fig. 5 and Fig. 6 which show the training and validation accuracy and loss curves respectively.

As shown in Fig. 5, the training accuracy increases rapidly during the initial epochs, reaching above 90% within the first few epochs and gradually converges toward approximately 99-100% by the later stages of training. The validation accuracy follows a similar trend, that stabilizes around 97-99% with minor fluctuations. The close alignment between training and validation accuracy curves indicates good generalization and absence of significant overfitting.

The loss curves in Fig. 6 shows that as the learning progresses, the training loss decreases steadily and approaches near-zero values. The validation loss remains low and stable throughout the training and shows no upward trajectory. Minor fluctuations at certain epochs reflect natural variability due to the imbalanced data distribution. However, the overall convergence pattern suggests stable training and effective learning of discriminative features. The absence of divergence between the training and validation curves across 50 epochs indicates stable

convergence and suggests that the model does not rely on memorization despite the relatively small dataset size.

4.3 Comparison with Existing Approaches

A comparative evaluation with deep learning architectures used in [20] which includes VGG-based and ResNet-based models is presented in Table 1. While VGG16 [16] and ResNet50 [4] models achieve accuracies of 89.64% and 98.41% respectively, the proposed model outperforms them with an accuracy improvement of 9.16% over VGG16 and 0.39% over ResNet50.

Computational Efficiency: In addition to classification performance, computational efficiency is evaluated to assess the suitability of the proposed method for real-world OCR preprocessing scenarios. The efficiency comparison is summarized in Table 1. Our model achieves an average inference time of 2.652 ms per word image, which is significantly lower than VGG16 (15.779 ms) and ResNet50 (14.711 ms). This corresponds to an inference time reduction of approximately 83.2% compared to VGG16 and 81.9% compared to ResNet50. Furthermore, the total training time for StrikeCNN

Table 1. Performance Comparison with benchmark models on Proposed Dataset

Model	Epochs	Accuracy	Precision	Recall	F1 Score	Inference Time (in ms)	AUC	Total Training Time (in s)	Model Size (in MB)
Resnet50 [4] [20]	50	0.9841	0.9724	0.9724	0.9724	14.711	0.9986	6455.10	89.68
VGG16 [16] [20]	50	0.8964	0.8143	0.9372	0.8525	15.779	0.9982	6820.24	72.13
Proposed	50	0.9880	0.9929	0.9659	0.9788	2.652	0.9987	844.05	16.36

is 844.05 s which is considerably lower than that of VGG16 (6820.24 s) and ResNet50 (6455.10 s).

The proposed model also requires substantially less storage with a model size of 16.36 MB compared to 72.13 MB for VGG16 and 89.68 MB for ResNet50. These reductions in inference time, training time and model size highlight the efficiency of StrikeCNN. At the same time, the model maintains strong classification performance. This makes the proposed approach well suited for large-scale document analysis systems and real-time OCR preprocessing pipelines, where computational resources and latency are critical constraints.

4.4 Discussion

The experimental results demonstrate that the proposed method effectively captures discriminative stroke and strike-through characteristics in handwritten word images. The high F1-score of 97.88% and AUC of 0.9987 indicate robust performance even under challenging handwriting variations and class imbalance conditions. Fig. 7 shows a screenshot of the original words present which consists of both valid and strike-out words.

Fig. 8 shows a screenshot of the words after applying our proposed model which shows that the final words are only the valid words without the strike-out words.

Compared with prior works that primarily report F-measure for strike-out detection and removal tasks, the proposed approach provides competitive performance using standard classification metrics while achieving lower inference time. These characteristics make it particularly suitable as a lightweight preprocessing module for OCR and handwritten document understanding systems.

Unlike generative or reconstruction-based approaches that attempt to recover struck-out content, the proposed method focuses on early-stage exclusion of invalid words, which simplifies system design and avoids unnecessary computational overhead.

A limitation of the current approach is that words with very light or partial strike-through patterns may occasionally resemble valid handwriting, which can affect classification confidence. However, such cases are rare and can be further addressed through larger datasets or adaptive thresholding strategies.

5 Conclusion and Future Work

This work presented a word-level approach for handling strike-out text in handwritten documents through binary classification. A lightweight convolutional neural network, referred to as StrikeCNN was designed to distinguish strike-out and non-strike-out word images using binarized inputs. By learning features directly from word images, the proposed method avoids reliance on handcrafted heuristics and eliminates the need for pixel-level strike removal. Experimental results on the proposed dataset demonstrate that the model achieves accuracy of 98.8%, while maintaining significantly lower inference time, training time and model size compared to commonly used pretrained architectures. These findings indicate that task-specific, compact CNN models can effectively address strike-out detection without the computational overhead associated with deeper networks. The proposed approach is therefore suitable for integration as a preprocessing or filtering module in handwritten OCR pipelines.

In the near future, more works on extending StrikeCNN to handle more complex correction

patterns, such as partially struck or overwritten words that are common in real-world handwritten documents. The robustness of the model can be further improved by evaluating it on larger and more diverse datasets encompassing multiple writing styles and document conditions. The quality of the model can also be enhanced by testing it on bigger and more varied datasets including various condition of documents and styles of writing. Exploring multilingual handwritten scripts gives a significant and promising direction for future research. Finally, the model can be integrated into a complete end-to-end OCR system in order to explore its impact on the overall recognition accuracy.

References

1. **Bhattacharya, N., Frinken, V., Pal, U., Roy, P. P. (2015).** Overwriting repetition and crossing-out detection in online handwritten text. 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), IEEE, pp. 680–684. DOI: 10.1109/ACPR.2015.7486589.
2. **Chaudhuri, B. B., Adak, C. (2017).** An approach for detecting and cleaning of struck-out handwritten text. Pattern Recognition, Vol. 61, pp. 282–294. DOI: 10.1016/j.patcog.2016.07.032.
3. **Dheemanth Urs, R., Chethan, H. (2021).** A study on identification and cleaning of struck-out words in handwritten documents. In Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2020. Springer, pp. 87–95. DOI: 10.1007/978-981-15-8530-2_6.
4. **He, K., Zhang, X., Ren, S., Sun, J. (2015).** Deep residual learning for image recognition. DOI: 10.1109/CVPR.2016.90.
5. **Heil, R., Vats, E., Hast, A. (2021).** Strikethrough removal from handwritten words using CycleGANs. International Conference on Document Analysis and Recognition, Springer, pp. 572–586. DOI: 10.1007/978-3-030-86337-1_38.
6. **Khargharia, D., Borgohain, S. K. (2024).** An aid to Assamese language processing by constructing an offline Assamese handwritten dataset. Proceedings of the 21st International Conference on Natural Language Processing (ICON), pp. 86–93.
7. **Nigam, S., Behera, A. P., Gogoi, M., Verma, S., Nagabhushan, P. (2023).** Strike off removal in Indic scripts with transfer learning. Neural Computing and Applications, Vol. 35, No. 17, pp. 12927–12943. DOI: 10.1007/s00521-023-08433-z.
8. **Nigam, S., Behera, A. P., Verma, S., Nagabhushan, P. (2024).** Deformity removal from handwritten text documents using variable cycle GAN. International Journal on Document Analysis and Recognition (IJDAR), Vol. 27, No. 4, pp. 615–627. DOI: 10.1007/s10032-024-00466-x.
9. **Nisa, H. (2024).** Off-line handwritten text recognition with cross-outs and corrections using deep learning. Ph.D. thesis, RMIT University. DOI: 10.25439/rmt.27599955.
10. **Nisa, H., Thom, J. A., Ciesielski, V., Tennakoon, R. (2019).** A deep learning approach to handwritten text recognition in the presence of struck-out text. 2019 international conference on image and vision computing New Zealand (IVCNZ), IEEE, pp. 1–6. DOI: 10.1109/IVCNZ48456.2019.8961024.
11. **Plamondon, R., Srihari, S. N. (2002).** Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on pattern analysis and machine intelligence, Vol. 22, No. 1, pp. 63–84. DOI: 10.1109/34.824821.
12. **Poddar, A., Chakraborty, A., Mukhopadhyay, J., Biswas, P. K. (2021).** Detection and localisation of struck-out-strokes in handwritten manuscripts. International Conference on Document Analysis and Recognition, Springer, pp. 98–112. DOI: 10.1007/978-3-030-86159-9_7.

13. **Purohit, A., Chauhan, S. S. (2016).** A literature survey on handwritten character recognition. *International Journal of Computer Science and Information Technologies (IJCSIT)*, Vol. 7, No. 1, pp. 1–5.
14. **Shivakumara, P., Jain, T., Pal, U., Surana, N., Antonacopoulos, A., Lu, T. (2022).** Text line segmentation from struck-out handwritten document images. *Expert Systems with Applications*, Vol. 210, pp. 118266. DOI: 10.1016/j.eswa.2022.118266.
15. **Shonenkov, A. V., Karachev, D. K., Novopoltsev, M. Y., Potanin, M. S., Dimitrov, D. V., Chertok, A. V. (2022).** Handwritten text generation and strikethrough characters augmentation. *Journal of Computer Optics*, Vol. 46, No. 3, pp. 455–464. DOI: 10.18287/2412-6179-CO-1049.
16. **Simonyan, K., Zisserman, A. (2014).** Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. DOI: 10.48550/arXiv.1409.1556.
17. **SK, M. S., Reddy, M. D. S., Gadar, M. P. M. (2024).** Revolutionary methodology for detecting and eliminating strike-through text in handwritten Kannada documents. *Journal of Engineering and Technology Management*, pp. 569–584.
18. **Sonkusare, M., Sahu, N. (2016).** A survey on handwritten character recognition (HCR) techniques for English alphabets. *Adv Vis Comput Int J*, Vol. 3, No. 1, pp. 1–12. DOI: 10.5121/avc.2016.3101.
19. **Tensmeyer, C., Martinez, T. (2017).** Analysis of convolutional neural networks for document image classification. 2017 14th IAPR international conference on document analysis and recognition (ICDAR), IEEE, Vol. 1, pp. 388–393. DOI: 10.1109/ICDAR.2017.71.
20. **Zhong, D., Palaiahnakote, S., Pal, U., Lu, Y. (2025).** Struck-out handwritten word detection and restoration for automatic descriptive answer evaluation. *Signal Processing: Image Communication*, Vol. 130, pp. 117214. DOI: 10.1016/j.image.2024.117214.

Article received on 07/02/2026; accepted on 03/04/2026.

**Corresponding author is Debabrata Khargharia.*