

MarkMatch: Same-Hand Stuffing Detection

Fei Zhao*, Runlin Zhang[†], Chengcui Zhang*, Nitesh Saxena[‡]

*Department of Computer Science, The University of Alabama at Birmingham, Birmingham, USA

[†]Department of Computer Science, University of Waterloo, Waterloo, Canada

[‡]Department of Computer Science and Engineering, Texas A&M University, College Station, USA

larry5@uab.edu, r496zhan@uwaterloo.ca, czhang02@uab.edu, nsaxena@tamu.edu

Abstract—We present *MarkMatch*, a retrieval system for detecting whether two paper ballot marks were filled by the same hand. Unlike the previous SOTA method *BubbleSig*, which used binary classification on isolated mark pairs, *MarkMatch* ranks stylistic similarity between a query mark and a mark in the database using contrastive learning. Our model is trained with a dense batch similarity matrix and a dual loss objective. Each sample is contrasted against many negatives within each batch, enabling the model to learn subtle handwriting difference and improve generalization under handwriting variation and visual noise, while diagonal supervision reinforces high confidence on true matches. The model achieves an F1 score of 0.943, surpassing *BubbleSig*’s best performance. *MarkMatch* also integrates Segment Anything Model for flexible mark extraction via box- or point-based prompts. The system offers election auditors a practical tool for visual, non-biometric investigation of suspicious ballots.

Index Terms—Ballot Stuffing Detection, Deep Learning

I. INTRODUCTION

Hand-marked paper ballots are a transparent and verifiable choice in many elections. However, this format remains vulnerable to same-hand stuffing attacks, where multiple marks are filled by the same individual to manipulate outcomes. For example, Barry Morphew cast a ballot for his missing wife [1], raising concerns about election integrity. Detecting such same-hand mark patterns is challenging, especially when the marks are small, sparse, and visually subtle.

BubbleSig [2], our previous work and also the first method in this field, used a Siamese neural network to compare an isolated pair of ballot marks and predict whether they were drawn by the same person. However, the training is based solely on a single mark pair, without considering alternative candidates or broader context. This makes the network susceptible to intra-writer variability and prone to false positives in ambiguous cases where a mark resembles multiple drawing styles. Moreover, its binary classification setup optimized the comparison of a mark pair in isolation, focusing only on pairwise match probability without considering how a given mark relates to other marks. As a result, the model lacks the ability to learn relative similarity across multiple samples and therefore may struggle to find the correct match among visually similar alternatives.

To overcome these limitations, we propose *MarkMatch*, taking a query mark and retrieving the most similar marks from a mark candidate pool using contrastive learning. Our contrastive model is trained with a dense similarity matrix

and a dual loss objective. Each sample is contrasted against all others (row-wise and column-wise) in the batch, allowing the model to capture fine-grained handwriting variation while enforcing high confidence on true matches through diagonal supervision (see Section II). This design improves robustness to handwriting variability, ambiguous strokes, and visual noise, yielding more reliable decisions in real-world ballot settings.

In addition, *MarkMatch* integrates the Segment Anything Model (SAM) [3] to support prompt-based mark segmentation using bounding boxes or point clicks. Unlike the method in [4], which relies on computationally intensive alignment with blank ballot templates, SAM enables efficient, direct extraction of mark segments from marked ballots alone. This allows election officials to easily extract a query mark segment and retrieve the top- k most similar instances from a mark database, offering a practical, non-biometric tool for investigating suspicious same-hand stuffing patterns.

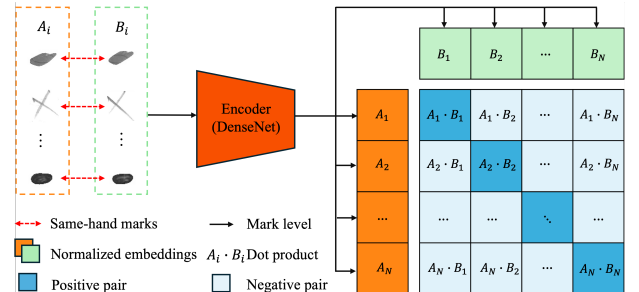


Fig. 1. The proposed contrastive learning model (detailed in Section II)

II. THE MODEL DESIGN

As illustrated in Fig. 1, we propose a contrastive learning model trained to distinguish positive pairs from a large set of negatives in each batch. Each ballot mark image is independently passed through a shared DenseNet121 encoder to obtain a fixed-length embedding. A training batch is constructed by sampling two aligned sets of ballot marks, A_i and B_j , for $i, j = 1, \dots, n$, such that each (A_i, B_i) corresponds to the same writer and is a positive pair, while each $(A_i, B_j, i \neq j)$ is a negative pair. Let $f(\cdot)$ denote the shared encoder. All embeddings are L2-normalized, and a similarity matrix $S \in \mathbb{R}^{n \times n}$ is computed using scaled dot product: $S_{ij} = \langle f(A_i), f(B_j) \rangle / \tau$, where τ is a temperature scaling factor set as 0.07. We collect softmax-based cross-entropy losses in both row and column directions. This encourages each embedding to align closely with its true positive counterpart while minimizing similarity

to other negatives within the batch. We further apply binary cross-entropy to sigmoid-activated diagonal scores, enhancing the model’s certainty on positive pairs. The loss is defined as:

$$\mathcal{L} = \frac{1}{2} (\mathcal{L}_{CE}^{\text{row}} + \mathcal{L}_{CE}^{\text{col}}) + \alpha \cdot \mathcal{L}_{BCE}$$

where $\mathcal{L}_{CE}^{\text{row}}$ denotes the row-wise cross-entropy (CE) loss computed as $\frac{1}{n} \sum_{i=1}^n \text{CE}(\text{softmax}(S_{i,:}), Y_{i,:})$, and $\mathcal{L}_{CE}^{\text{col}}$ is its column-wise counterpart $\frac{1}{n} \sum_{j=1}^n \text{CE}(\text{softmax}(S_{:,j}), Y_{:,j})$. The ground truth matrix $Y \in \{0, 1\}^{n \times n}$ assigns label 1 to diagonal entries (positive pairs) and 0 to all off-diagonal entries (negative pairs). \mathcal{L}_{BCE} represents the binary cross-entropy (BCE) loss applied to the diagonal entries: $\frac{1}{n} \sum_{i=1}^n \text{BCE}(\sigma(S_{ii}), Y_{ii})$, where $\sigma(\cdot)$ is the sigmoid function. The coefficient α is set to 1 based on empirical tuning. We train and evaluate our model using the same large-scale dataset and the configuration as *BubbleSig* [2]. As shown in Fig. 2, our model achieves an F1 score of 0.943.

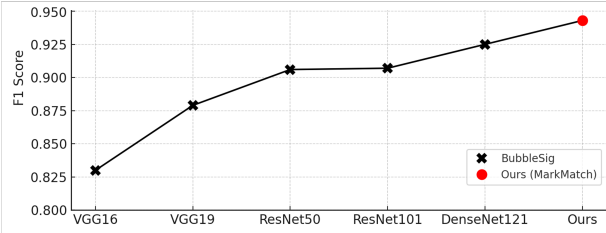


Fig. 2. F1 comparison between *BubbleSig* [2] variants and our model

For deployment, *MarkMatch* adopts a retrieval-based formulation. Given a query mark, the model computes similarity scores against a pool of candidates and applies a softmax over these scores to obtain a normalized ranking. The top-ranked marks are returned as the most likely same-hand instances. This formulation aligns naturally with contrastive learning and yields more reliable predictions than binary classification on isolated pairs, particularly in visually ambiguous cases.

III. DEMONSTRATION FEATURES

MarkMatch provides an interactive platform for analyzing and comparing hand-drawn ballot marks through visual similarity. The system begins by using SAM to extract individual mark segments from a scanned ballot image (Step 1 in Fig. 3). Users can interactively select regions using bounding boxes or point clicks, allowing online and flexible segmentation across varying mark shapes and ballot layouts. Once segmented, each selected mark can serve as a query, which is compared against a pool of candidate marks. For every query, the system computes a similarity score with every mark in the pool using our contrastive model. These results are visualized through two components, shown as Step 2 in Fig. 3:

- **Ranking Table:** For each query mark, the system displays the top-5 retrieved pool marks ranked by softmax-normalized similarity scores. For each match, the table includes the candidate’s rank, filename, image, softmax score, and raw similarity logit. This view allows auditors to assess both the statistical evidence and visual resemblance behind each prediction.

- **Heatmap Visualization:** A matrix showing softmax-normalized similarity scores between candidate pool marks (rows) and query marks (columns). Each cell reflects the model’s confidence that a pool mark matches the corresponding query. This global view reveals retrieval confidence distributions across all pairs.

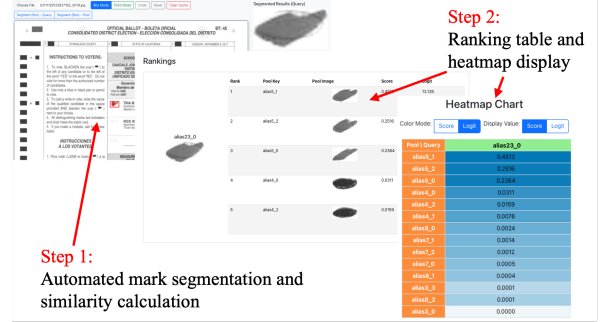


Fig. 3. *MarkMatch* interface. The system segments marks using prompt-based inputs (box or point), computes similarity between the query (green) and candidates (orange), and visualizes softmax-normalized scores via heatmap and ranking. To preserve privacy, marks are anonymized using letter-number aliases (e.g., alias5_0). In this example, three top-ranked matches from the same ballot (alias5) are retrieved for the query mark alias23_0, and manual review confirms both ballots were filled by the same individual.

Together, these features support transparent, scalable analysis of handwriting consistency across ballots. The web interface provides real-time feedback, visual overlays, and ranked comparisons, enabling auditors to efficiently investigate potential same-hand stuffing frauds in real-world audit workflows. A demonstration video is included in the supplementary material.

IV. CONCLUSIONS & ACKNOWLEDGEMENT

We presented *MarkMatch*, a retrieval-based system for detecting same-hand ballot marks using contrastive learning. By modeling relative similarity across batches, it improves robustness in ambiguous and noisy cases. *MarkMatch* integrates SAM for prompt-based segmentation on the fly and provides intuitive visualizations via heatmaps and ranked tables, offering election auditors an efficient, non-biometric tool for scalable, transparent ballot review. This work was supported by NSF CNS-2154589 and 2154507, “Collaborative Research: SaTC: CORE: Medium: Bubble Aid: Assistive AI to Improve the Robustness and Security of Reading Hand-Marked Ballots,” \$1,200,000, 10/01/2022-09/30/2026.

REFERENCES

- [1] Neil Vigdor, “Colorado man pleads guilty to casting missing wife’s ballot for trump,” <https://www.nytimes.com/2022/07/22/us/politics/suzanne-morphew-ballot-trump.html>, 2022, Accessed: 2024-02-07.
- [2] Fei Zhao, Chengcui Zhang, Maya Shah, and Nitesh Saxena, “Bubblesig: Same-hand ballot stuffing detection,” in *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2024, pp. 507–510.
- [3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al., “Segment anything,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.
- [4] Fei Zhao, Chengcui Zhang, Nitesh Saxena, Dan Wallach, and Akm Shahriar Azad Rabby, “Ballot tabulation using deep learning,” in *2023 IEEE 24th International Conference on Information Reuse and Integration for Data Science (IRI)*. IEEE, 2023, pp. 107–114.