Ballot Tabulation Using Deep Learning

Fei Zhao
Dept. of Computer Science
The University of Alabama at Birmingham
larry5@uab.edu

Chengcui Zhang
Dept. of Computer Science
The University of Alabama at Birmingham
czhang02@uab.edu

Nitesh Saxena
Dept. of Computer Science & Engineering
Texas A&M University
nsaxena@tamu.edu

Dan Wallach
Dept. of Computer Science
Rice University
dwallach@cs.rice.edu

AKM Shahariar Azad Rabby
Dept. of Computer Science
The University of Alabama at Birmingham
arabby@uab.edu

***Abstract*—Currently deployed election systems that scan and process hand-marked ballots are not sophisticated enough to handle marks insufficiently filled in (e.g., partially filled-in), improper marks (e.g., using check marks or crosses instead of filling in bubbles), or marks outside of bubbles, other than setting a threshold to detect whether the pixels inside bubbles are dark and dense enough to be counted as a vote. The current works along this line are still largely limited by their degree of automation and require substantial manpower for annotation and adjudication. In this study, we propose a highly automated deep learning (DL) mark segmentation model-based ballot tabulation assistant able to accurately identify legitimate ballot marks. For comparison purposes, a highly customized traditional computer vision (T-CV) mark segmentation-based method has also been developed to compare with the DL-based tabulator, with a detailed discussion included. Our experiments conducted on two real election datasets achieved the highest accuracy of 99.984% on ballot tabulation. In order to further enhance our DL model's capability of detecting the marks that are underrepresented in training datasets, e.g., insufficiently or improperly filled marks, we propose a Siamese network architecture that enables our DL model to exploit the contrasting features between a hand-marked ballot image and its corresponding blank template image to detect marks. Without the need for extra data collection, by incorporating this novel network architecture, our DL model-based tabulation method not only achieved a higher accuracy score but also substantially reduced the overall false negative rate.**

***Index Terms*—Deep learning, Ballot tabulation, Computer vision**

I. Introduction and Related Work

Scanned images of hand-marked paper ballots have been used to analyze, verify, and independently recount ballots (tabulation), either manually or using traditional computer vision (T-CV) methods coupled with common data mining techniques such as clustering and/or traditional machine learning techniques [1, 2]. Many factors could affect the accuracy of T-CV methods, including marginal marks (e.g., checks and crosses), marks outside of the voting bubble/box, and scanning errors (e.g., caused by stains and creases on paper ballots), etc. Taking marginal marks as an example, most existing optical scan systems will either miss or misinterpret them using a predefined pixel intensity threshold, because the average pixel intensity of these marginal marks falls between that of a fully marked and an empty marked voting target. Marks outside of the voting target area will be largely missed by optical scan systems that are not configured to look at marks outside of the voting target areas (bubbles/boxes). The proposed work is intended to be a highly-automated and more accurate ballot tabulation assistive tool using computer vision (CV) and artificial intelligence (AI) techniques, which is the first integrated framework of its kind to assist the tabulation of paper ballot at a high level of robustness and integration never achieved before.

Several studies have endeavored to enhance both the automation and accuracy of ballot tabulation, primarily using CV techniques: the operator-assisted tabulation system proposed in [1] and [3] provides a user interface to expedite ballot auditing. However, this method requires an operator to manually annotate the voting target area, locate names of candidates, and set a threshold of pixel intensity in order to classify a voting target as marked or unmarked. This manual configuration is required for each distinct contest, requiring substantial manual intervention. There are some other works aiming to improve the automation of this process. For example, the work in [4] focuses on automated mark segmentation, which is a necessary step in automated tabulation. They evaluated the absolute differencing technique both with and without adaptive thresholded images, by comparing marked ballots to an unmarked one. The technique with adaptive thresholding gave the best detection rates for marks, but with an increase of false positive rate. However, their experiments were very limited, based on 4 synthetic ballot images **synthetically** filled by some algorithm. Xiu et al. [5] proposed a method to detect marks collectively within a ballot instead of an isolated fashion, assuming consistency in the same voter's marking style. The classification method was built based on a traditional data mining technique Modified Quadratic Discriminant Functions [6], and the 300 features were generated using 2D Fast Fourier Transform. This method is able to classify three different marks, including check marks, "X" marks, and filled marks. Again the method cannot automatically segment marks, and the hand-built test dataset is relatively small with 730 marks. In [2], a model was built to classify marks into 7 classes, i.e., empty marks, filled

This work was supported by NSF CNS-2154589, 2154443, 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.

marks, and five types of marginal marks consisting of checkmark, cross, partially filled, overfilled, and lightly filled. 55 commonly used computer vision features and 9 **hand-crafted** image features targeting marginal marks were used to train a few off-the-shelf traditional machine learning models, and the 9 customized features yielded the highest classification accuracy of 94%. All the three above methods fall into the category of T-CV and traditional machine learning/data mining techniques, in which features are manually crafted and selected rather than learned, and the models are fixed structures such as Support Vector Machines, Decision Trees, and Simple Logistic Regression. None of them attempted to build a highly automated ballot tabulation tool. Similarly, Barretto et al. [7] trained a Convolutional Neural Network to classify various styles of marks extracted from a ballot image dataset and achieved the state-of-the-art (SOTA) prediction performance. However, this work requires manually annotating voting target areas from ballot images, and their high accuracy largely depends on the fact that all marks receiving a classification confidence score lower than a pre-fixed threshold (95%) are subject to manual inspection and thus will be counted as correctly classified.

As reviewed above, so far, there is no AI-assisted highly automated and efficient solution to ballot tabulation for scanned hand-marked paper ballots.

Compared to T-CV techniques, DL does not need human-guided feature extraction or manual feature selection. Also, it is not uncommon to use hundreds of features in T-CV; the extraction of such features can be very time-consuming and is not easily parallelizable. In contrast, DL fully automates the feature extraction by assigning credits/contributions to hundreds of thousands of neurons through many layers of neural networks. As a result, the deep image features learned through DL are often more resistant to variations and noise/artifacts (e.g., creases) and more generalizable to unseen data. In addition, DL models are typically more flexible and re-trainable for new domain/dataset [8], compared to highly customized CV algorithms for specific domains. Currently, DL is under-explored in the field of voting system improvement.

The main contributions of this paper are as follows:

- A T-CV-based and a DL-based mark segmentation models are proposed, and both are highly automated and the first in their respective kind, with a good generalizability to various types of ballot without major modifications. Compared to the proposed T-CV-based model, the DL-based model does not require fine-level image registration, cutting down the computation drastically. We developed the T-CV-based mark segmentation model for the following purposes: 1) for accuracy and performance comparison with the DL-based mark segmentation model; 2) for automatically obtaining a large training dataset for training the DL-based model since it is very costly to manually collect ground truth of mark segments. As demonstrated in our experimental results, the DL-based mark segmentation model is much more accurate than the T-CV-based model.
- We further propose a Siamese network architecture, which allows our DL-based model to utilize the contrasting features between a hand-marked ballot image and its corresponding blank template ballot image, effectively reducing false negatives resulting from lightly/insufficiently filled, and/or improperly filled marks underrepresented in the training dataset. Importantly, this strategy mitigates the need of collecting extra training data for such underrepresented marks.
- For image registration in ballot tabulation, we propose the first fully automated coarse-to-fine-level image registration framework. Compared to the state-of-the-art ballot registration methods, the proposed method does not require any assumptions on the ballot layout, and can be easily generalized to different ballot layouts.

In summary, current election systems deploying basic ballot scanners suffer from inaccurate tabulation problems. In this work, we take advantage of the advances in AI and CV techniques, as well as large ballot datasets, and aim to significantly improve both the robustness and accuracy of hand-marked ballot scanning.

II. Problem Definition

This study is restricted to scanned mail-in paper ballots and focuses on the mark segmentation and tabulation of such. Any signature or write-in content of voters on ballots are beyond the scope of this study. No historical or biometric data of voters are required by this study. There are three main tasks addressed in this paper: Ballot Image Registration, Target Area Localization, and Mark Detection and Segmentation.

Ballot Image Registration: Although the same camera angle and distance to the paper ballots are assumed during scanning, imperfect scans can happen, e.g., the paper ballot was not completely flattened out or not placed in a position perfectly aligned with the camera, which could substantially affect the performance of the subsequent modules such as voting target area localization and mark segmentation. Fig. 1 is the superimposition of a marked ballot image over its corresponding blank template ballot image, showing a lot of misalignments. These misalignments can have a significant negative impact on the downstream tasks. Therefore, ballot image registration needs to be performed.

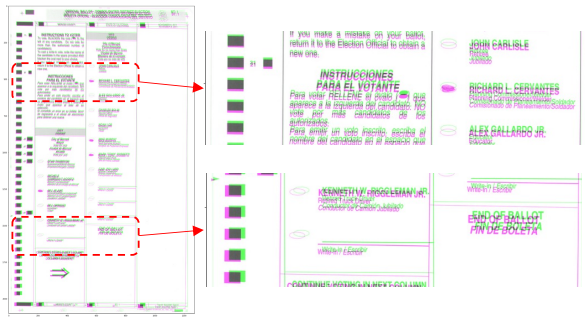

Fig. 1: Misalignment between a ballot image and its corresponding blank ballot template image

Target Area Localization: In Fig. 2, the red bounding boxes are our proposed voting target area, which has a different definition than that of the voting target area commonly used in the SOTA methods. All SOTA methods, require operators to manually annotate/extract the voting target area for each candidate on the ballot template. The labeled target areas are shown as orange boxes in Fig. 2(a). This manual process can be time-consuming and imprecise. Moreover, in filled ballots, there can be marks outside of the labeled target areas that are bound to be missed due to this old definition of target area. Many states have a definition of "intent of the voter", which is to say, how a machine might interpret a mark is not the final word. If human inspectors can intuit the mind of the voter enough to identify their intent, then that is the proper and correct interpretation of the ballot, such as the out-of-box cross marks in Fig. 2(c) that do indicate votes for the two candidates but remain undetectable by existing methods. By expanding our field of view beyond the marks inside bubbles, we can potentially better capture intent of the voters who do not follow the instructions. In this paper, we propose a new definition of voting target areas, shown as red boxes in Fig. 2(b), detected as voting "cells" in a structured ballot layout that include not only the marking areas, but also the candidate names/options, and the surrounding background area as well. By adopting this new definition, the proposed ballot tabulation method can be more robust in handling marks outside of bubbles/boxes (Fig. 2(c)). Furthermore, we proposed a highly automated voting target area localization method which requires minimum manual aid.

(a) (b) (c)

Fig. 2: Voting target areas in (a) and (b) are based on the old and new definitions, respectively. In (c), the marks are still in the newly defined voting target areas.

Mark Detection and Segmentation: The SOTA method proposed in [1] requires operators to manually set a threshold of pixel intensity in order to classify a manually labeled voting target as marked or unmarked. In this paper, we treat this task as an object semantic segmentation problem where each pixel of the voting target area will be assigned a label as either a mark or a non-mark pixel. We propose two independent automated models: a T-CV model and a DL mark segmentation model, to obtain the mark segments (including marginal and out-of-box ones) from the proposed voting target areas. An example of the expected result of this process is shown in Fig. 3.

Fig. 3: An example of an obtained mark and its segment

III. Methodology

A. Ballot Image Registration

Ideally, the scanned paper ballots of one election are supposed to be well aligned. In practice, however, most scanned paper ballots are not well aligned, e.g., the severe misalignment in Fig. 1.

Based on observations on real-world datasets, there are often rotations and shifts among ballot scans. The misalignment can have a significant negative impact on the performance of the subsequent mark segmentation module, especially in the T-CV mark segmentation model. One of the SOTA ballot registration methods ([9]) manually selected and annotated the 4 black boxes on the 4 corners of the ballot as references. The authors then use an affine transformation to align the scanned paper ballots. However, this method assumes that the ballot must contain at least 4 black boxes, one in each corner, which limits the generalization of this method. Furthermore, the authors have not tested these methods on any other real-world dataset. Wang et al. [1] proposed an alignment method based on linear Hough Transform, in which they assume that a ballot must contain two or more sufficiently long vertical or horizontal lines. Furthermore, this method adopts a local alignment method, which requires operators to manually annotate several important areas (the areas containing election contests) in the ballot image and crop them out as sub-images. Then, the authors align sub-images in the downstream tasks. In the computer vision society, the most popular image registration method is feature-based alignment, in which a set of image feature points such as SIFT (Scale Invariant Feature Transform [10]) are extracted, and one image is warped into the template image by the calculated transformation matrix so that the feature points in both images line up to the maximum extent. However, using this method alone is insufficient for our ballot image registration task as there are too many noisy features. To address the misalignment issue, we propose a two-step fully automated ballot image registration method based on CV techniques. In the proposed method, no human intervention, such as manually choosing reference areas or annotating the ballots, is required.

The general idea of our method is to align two ballots (a blank template ballot image and a marked ballot image) progressively from a coarse level to a finer level. For the coarse level alignment, we adopt the state-of-the-art feature-based image registration method ORB [11]. ORB is rotation invariant and resistant to noise, which is perfectly suitable for coarse level alignment. After applying ORB, the misalignment caused by rotation and shift can be largely fixed. However, the misalignment caused by other nonrigid transformations still remain. Therefore, an optical-flow-based fine-level alignment method is adopted [12–14]. In this step, instead of warping the coarsely aligned marked ballot to the template image, we warp in the other direction (from template to the marked ballot), to avoid excessive distortion to the ballot image due to one more round of transformation. The mean norm of the estimated optical flow vector for each ballot is also used to find the

ballots that have a significantly different layout (e.g., due to scanner errors) than the template. If any such outlier is found, the corresponding ballot will be subject to further adjudication. It is worth noting that this expensive fine-level registration is necessary in traditional CV-based mark detection since slight misalignment could lead to significant noise, but not needed in the DL for which the low-cost coarse-level registration is all that is needed.

B. Target Area Localization

The definition of a voting target area is provided in Section II, and an example is shown in Fig. 2. In an optical scan voting system, voters choose by filling a bubble (Fig. 4(a)) or by connecting an arrow (Fig. 4(b)) on the printed ballot next to their chosen candidate. Such bubbles and arrows are voting objects of interest that we want to detect and match, in order to locate voting target areas.

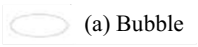

Fig. 4: Voting objects of interest

Ballot templates can have different layouts and sizes for different elections, so can the voting objects of interest. To generalize our algorithm, we resized blank template ballot images to certain size scale (e.g., using the blank template of one election as the size scale reference) while preserving the aspect ratio so that the size and shape of voting object of interest in all blank template ballot images fall into a similar scale, facilitating the subsequent pattern matching. In this project, we use a template matching algorithm to locate all the voting objects from a blank ballot template image.

In particular, we use an OpenCV function cv2.matchTemplate() for this purpose, which simply slides the voting object of interest over the blank template ballot image (as in 2D convolution) and compares the voting object of interest with each patch of blank template ballot image within the sliding window. Fig. 5(a) shows some examples of detected bubbles (confined within red bounding boxes).

This process stores all the location coordinates where it finds a match with the voting object of interest in the blank template ballot image. The sliding window may find multiple matches for one voting object of interest, and the coordinates for each best local match will be saved and then stored in a hash table as a (key, value) pair.

According to our observations on several large ballot image datasets, all ballot templates contain vertical dividing lines to separate the ballot into multiple columns. Detecting those vertical dividing lines can help determine the left & right boundaries of voting target areas. Although our method does not depend on vertical dividing lines, we provide a solution to detecting such as follows, followed by a more general solution to detecting (physical/virtual) column boundaries. To detect such lines we used probabilistic Hough transformation[15]. Hough transform is a popular technique to detect shapes such as lines if the shape can be represented in a mathematical form. It can detect a line even if it is slightly broken or distorted. To further reduce noise and irrelevant lines detected, we only consider detected vertical lines around voting object of interest. These lines will be used later to determine the left and right boundaries of each target voting area. Fig. 5(b) shows the detected vertical lines.

In order to locate the top and bottom boundaries of each voting target area, we need to first locate the dividing point between every two vertically adjacent voting objects (e.g., bubbles) in the same column (as confined by the column boundaries detected from the previous step). Given the coordinates of a detected voting object, its nearest voting object in the same column can be identified, and the vertical distance of those two can be calculated, as well as the middle point in between the two objects along the vertical direction. Next, the vertical distance between the two voting objects can be used as an estimate of the height of each voting target area, based on which the top and bottom boundaries can also be located, with reference to the middle point. Fig. 5(c) shows the located target voting areas, with their bounding boxes colored in red.

In case that vertical lines are not physically present, we can still detect those **virtual** dividing lines by first clustering detected voting objects based on their x-coordinates so that each group corresponds to a column, then, locating the virtual column dividing lines by using an approach similar to the one used to locate horizontal dividing lines.

After all the target areas are located from the template, they can be presented to the staff during the ballot configuration phase where the staff enters the candidate's name/option for each voting target area, once and for all (for each election). Then, each marked ballot is sent to a mark detection module - whenever a mark is detected within a target area, the corresponding candidate gets one more vote.

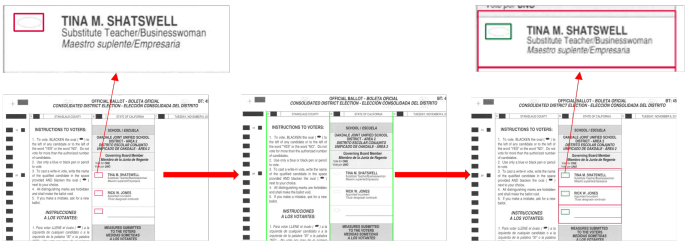

Fig. 5: (a) Detected voting objects of interest (bubbles), (b) detected vertical dividing lines, (c) located target voting areas

C. Mark Detection and Segmentation

The state-of-the-art mark detection methods are mainly based on pixel intensity thresholding. Wang et al. [1] manually selected and annotated the voting target area, and hand-picked a threshold of pixel intensity to classify the voting target as marked or unmarked. This process largely depends on the operator's personal experience and expertise. Furthermore, since scanners cannot guarantee the same lighting condition for all the paper ballots during scanning, the fixed threshold is not robust in practice. To detect marks from ballots, we

propose two independent models: traditional computer vision-based model and deep learning-based model.

1) Traditional CV-based Mark Segmentation: To obtain the marks from target ballots, we propose a T-CV-based mark segmentation model consisting of morphological transformation, denoising filter, adaptive binarization, and connected component labeling. The general idea of the method is to find the differences between the template ballot image and the target ballot image.

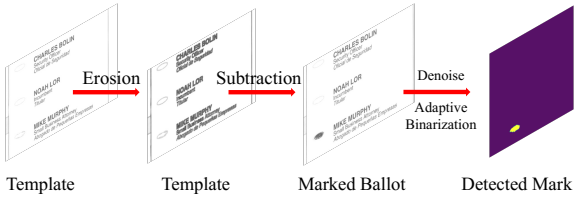

Fig. 6: Mark segmentation using traditional computer vision techniques

The proposed T-CV-based mark segmentation model (Fig. 6) relies on a pair of images, i.e., a blank template ballot and a marked ballot, which are well-aligned by the proposed image registration method. Well aligned template and target ballot images are first converted into grayscale images. We then apply element-wise subtraction on them, in which the marked ballot is subtracted from the blank template ballot. Since the marks on the marked ballot should be darker than the corresponding area on the blank template ballot, only the positive values in the difference matrix can potentially indicate the marked areas. Based on our observation, a slight misalignment between two images can make the difference matrix noisy. To improve the robustness, we apply a morphological operation on the template image before the subtraction, in which an erosion operator is adopted to expand dark outlines of bubbles (Fig. 6). The expanded area on the template ballot can effectively counter off the majority of positive values in the difference matrix caused by slight misalignment. Since only the areas with positive values are needed, we discard the negative values and clip them to 0. Next, as the difference matrix can be noisy, we adopt a median filter to denoise it. Then, we apply a common adaptive image binarization method, Otsu's Binarization [16], on the difference matrix, and segment the binarized difference matrix by using the Connected Components Labeling [17]. After all the above operations, the method is expected to detect the marks from ballots and their corresponding segments. The entire process is fully automated without any human intervention. It does not require the operator to manually define a fixed threshold or to annotate target voting areas to detect and segment marks.

2) Deep Learning-based Mark Segmentation: The SOTA model, Mask-RCNN (Regional Convolutional Neural Network) [18], utilizes a relatively simple method to achieve success in the task of object detection and instance segmentation. The proposed DL-based mark segmentation model is based on Mask R-CNN [18]. There are two classes in the proposed DL-based model, including the "background" and

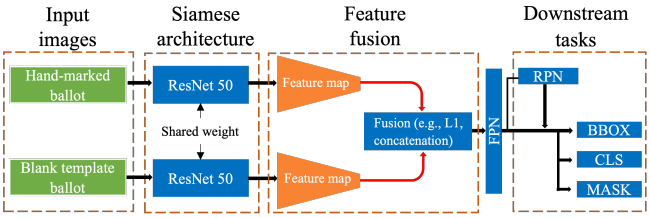

Fig. 7: The proposed Siamese architecture-based DL mark detection models

"mark" classes. A pixel classified as "background" indicates that the pixel is classified as unmarked, otherwise marked. However, a DL model usually requires a large amount of data since it trains by using original input data (vs engineered features) directly. There is no available public dataset with ground truth for the mark detection and segmentation task, and manually extracting the boundary of each mark from each ballot image can be intimidatingly costly. We created a training dataset the ground truth of which is obtained by using the proposed T-CV-based mark segmentation, eliminating the need for costly human annotation. The training dataset obtained this way is not expected to have perfect ground truth, however, we hypothesize that the deep learning model can still pick up the discriminative features for marks, ***even with some labeling errors*** in the training set.

In order to improve the model's ability to detecting the marks that are usually underrepresented in training dataset, such as those marginal or improperly filled marks, we extend our DL-based mark segmentation model with a Siamese network architecture, which enables the model to handle a pair of input images: a hand-marked ballot image and its blank template ballot image. As illustrated in Fig. 7, we propose two variations of the Siamese network architecture: Siamese$_{L1}$ model, where an element-wise L1 distance is computed between the feature maps extracted from the two input images, and Siamese$_{\oplus}$ model, where these feature maps are concatenated channel-wise. The Siamese$_{L1}$ model, by using the element-wise L1 distance upon the feature maps, aims to guide the model to focus on the discrepancies between the two input images. On the other hand, the Siamese$_{\oplus}$ model uses channel-wise concatenation to fuse the feature maps. Instead of explicitly instructing the model to focus on discrepancies, the Siamese$_{\oplus}$ model is designed to autonomously discover and leverage the relationship and interplay of the two feature maps during its training process. The fused features are then sent to feature pyramid network (FPN) and other Mask-RCNN's downstream networks, e.g., region proposal network (RPN), bounding box regression (BBOX) and classification (CLS) heads as well as segmentation (MASK) head. An evaluation of our proposed Siamese network architecture is provided in Section V-B.

IV. Dataset and Experiments

The experiments are conducted on two real-world ballot datasets: Stanislaus County and Merced County of Califor-

nia state. Each county has a blank template ballot image, with raw ballot scans unaligned. In the Stanislaus County dataset, there are 3,151 scanned ballot images with the resolution 1700x2800, containing 2,211 ballot images in training dataset, 470 ballot images in validation dataset, and 470 ballot images in test dataset. The Merced County dataset contains 7,120 scanned ballot images with a resolution of 1272x2100, divided into 180 for training (fine tuning), 20 for validation, and 6,920 for testing. In the real scenario, the layout of ballots used in different elections can be very different. According to our observations, the model trained using the Stanislaus dataset demonstrated decent generality when applied to Merced County data, although not as good. A common technique used in the DL field is fine-tuning, which is used to tune a pre-trained DL model using training dataset from the current dataset previously unseen by the pre-trained model. The hypothesis is that, only a relatively small dataset is needed for fine-tuning if the new dataset shares a similar nature with the original dataset, and retraining can be done much faster than that for the initial pre-trained model. Therefore, the training and validation datasets are relatively small. The tabulation ground truth is annotated by one expert and reviewed by three others. The total number of target areas of the testing set is 92,780. According to our experiments, this fine-tuning, while not costly, can significantly improve the segmentation accuracy. The information of all the datasets is shown in Table I.

TABLE I: Summary of datasets (# of marks)

Dataset	Train	Validation	Test
Stanislaus	13,266	2,820	2,820
Merced	2,340	260	89,960
Overall	15,606	3,080	92,780

The experiment of T-CV mark segmentation model-based tabulation is relatively straightforward. Since the proposed T-CV-based model does not involve any training, this tabulation method will be applied to the testing dataset directly. To be specific, all the ballot images in test dataset will be sent to the proposed ballot image registration process firstly. After the fine level alignment process, the well-aligned test samples will be sent to the T-CV-based mark segmentation model. For each test ballot, this method will generate a mark segmentation map. In order to accelerate the processing, we adopted parallel computing techniques in the T-CV-based model. Since all the computations are based on NumPy arrays, we utilize Joblib [19] to parallelize the pipeline. The experiment was run on 14 CPUs (2.4GHz Intel Xeon E5-2680).

For the proposed DL mark segmentation model-based tabulation, as neither Stanislaus nor Merced dataset provides the mark segmentation and tabulation ground truth, we apply our proposed T-CV-based mark segmentation model on the training and validation datasets to gain a reasonable approximation of the actual ground truth for segmentation. Since this study does not consider the write-in content (e.g., write-in candidates) on ballots, but the T-CV-based model is able to pick up write-in content together with marks, we need a way to remove the detected write-in content from the T-CV-based model's output so that it will not misguide the training of the DL-based mark segmentation model. This can be done by asking the user to provide a special tag ("write-in") for each write-in target area on the ballot template detected by the proposed voting target localization algorithm (Section III-B). Then anything detected from a "write-in" area will be removed from the training dataset. By using the above strategy, we do not need to hire experts to annotate hundreds of thousands of ballots in the training and validation datasets. In real scenarios, it is not realistic to train a DL-based mark segmentation model from scratch for each contest or different type of ballot. A more practical way would be to pre-train a DL-based mark segmentation model on the existing dataset. Then, for different contests or different types of ballots, we only need a few samples of new data to fine-tune the pre-trained model, in the hope that the fine-tuned model can fit the new data much better. Therefore, in this experiment, we first train a DL-based mark segmentation model on the Stanislaus County ballots. In this step, the model is trained on a single NVIDIA Tesla P100 16GB GPU with 100 epochs with an initial learning rate of 0.00001. Then, the pre-trained model is fine-tuned by using the 200 Merced County ballots.

Regarding evaluation metrics, we consider a voting target area as one sample. To classify the voting target area is straightforward: if there is a detected mark in a voting target area, this target area is classified as "marked" or "vote", otherwise "unmarked" or "non-vote". A False Positive target area (FP) means the ground-truth label of the area is "non-vote", but the predicted label is "vote". A False Negative target area (FN) means that the area should be classified as "vote", but the prediction is "non-vote". In this experiment, we use Accuracy (ACC) as the metric, with the following definition:

$$ACC = 1 - \frac{FP + FN}{the\ total\ number\ of\ target\ areas} \quad (1)$$

V. Results and Discussion

In the ballot tabulation experiment, the accuracies of our highly automated DL and T-CV model-based tabulation methods are 99.984 % and 99.921%, respectively (Table II). Only 1 ballot was manually adjudicated, which involves a scanning error (the same candidate is scanned twice, and ballot layout is changed) detected by our ballot image registration algorithm (Section III-A). We count this ballot as a correct prediction since it was successfully picked up and sent for adjudication.

TABLE II: The result of ballot tabulation experiment

Test Dataset		T-CV Model			DL Model		
Name	# of Marks	FP	FN	ACC	FP	FN	ACC
Stanislaus	2,820	5	0	99.823%	0	3	**99.894%**
Merced	89,960	68	0	99.924%	2	10	**99.987%**
Overall	92,780	73	0	99.921%	2	13	**99.984%**

A. Notable Cases of T-CV and DL

The proposed T-CV-based tabulation method detects marks by examining the difference between a marked ballot image and its corresponding blank template ballot image. Therefore, noise/stains/stray marks on ballots, shown in Fig. 8, could

lead to increased false positives. In Fig. 8(a), a gold-colored printing stain overlapping the bubble target of the candidate "CHARLES BOLIN", is detected as a valid mark based on the difference between the marked ballot and the corresponding blank template. In Fig. 8(b), there are three line-like small segments in the voting target area of "SYNTHIA L. JON". In this case, the voter probably wanted to erase marks but did not erase them completely. Therefore, they are detected by the differences between the ballot and the template. Similar to (b), Fig. 8(c) shows another noise case, in which there is a hand-drawn scratch line detected by the T-CV-based model. However, our proposed DL mark segmentation model-based tabulation method detects and segments marks based on the knowledge it learned from the training dataset. It does not need to calculate the differences between two inputs nor rely much on alignment. As illustrated in Fig. 8, neither the printing/scanning errors nor noise/stains/stray marks can fool our DL mark segmentation model.

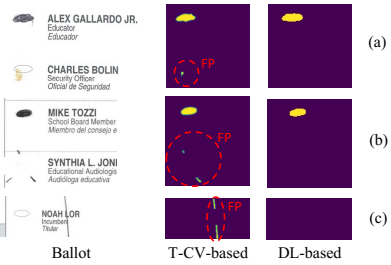

Fig. 8: Ballots with noise/stains/stray marks and the corresponding segmentation masks

To further evaluate the tabulator's robustness, we test both T-CV and DL model-based tabulation methods on several extreme cases, e.g., folded paper ballots and wrinkled paper ballots. In real scenarios, mail-in paper ballots could be folded in an envelope or wrinkled, as shown in Figs. 9 and 10. Despite slight distortions in lines and text due to folding, both T-CV and DL model-based methods work perfectly. However, the wrinkled ballots posed a challenge for the T-CV-based mark segmentation model, leading to numerous false positives. This is because the T-CV-based model highly depends on the quality of the alignment between two input images, and this type of severe misalignment is hard to be eliminated (Fig. 10). In contrast, the DL-based mark segmentation model delivered accurate predictions despite these challenges. It's evident that the proposed DL-based model is more robust to cases with severe misalignment, or noise/stains/stray marks, compared to the T-CV-based model.

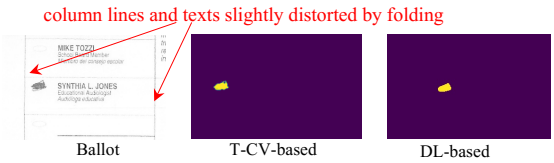

Fig. 9: Folded ballots and segmentation masks

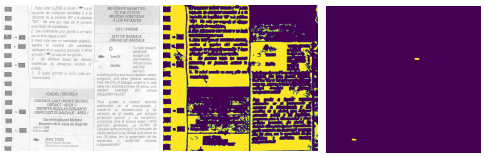

Fig. 10: A wrinkled ballot and its segmentation mask

B. DL with Siamese Network Architecture

The previous section shows that the DL-based mark segmentation model can be more robust and generalizable in cases of severe misalignment and noise/stains/stray marks. However, it does not mean that the DL-based model is without any limitations. In Fig. 11, we can see that the shape of the marks is quite different from that of a typical mark in the Stanislaus and Merced datasets. Fig. 11(b) visualizes the marks' segments correctly picked up by T-CV-based model. Our DL-based model was unable to detect any of these improper marks. This is because only one ballot in our dataset contains this type of marks, heavily underrepresented in the training dataset. In general, we can improve our DL-based model's performance on these kinds of marks by adding more similar ballots into the training dataset. However, data collection for such ballots would require additional human effort.

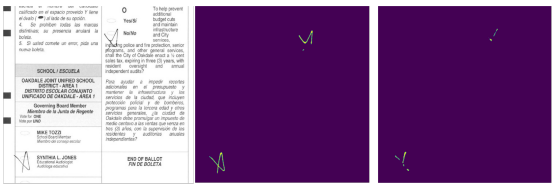

Fig. 11: Improper marks are correctly picked up by the T-CV-based model and the Siamese-based DL model. The DL-based model without Siamese architectures fails to detect this type of marks.

We propose a novel approach, introduced in Section III-C2, that integrates our DL-based mark segmentation model with Siamese network architectures to enhance the ability of detecting underrepresented mark types in the training dataset. This is accomplished by taking advantage of contrasting features between a marked ballot image and its blank template image. The two variants: Siamese$_{L1}$ and Siamese$_{\oplus}$, are tested on the same dataset and with the same training procedure as our DL-based model. As shown in Table III, by implementing our Siamese network architectures, not only has the tabulation accuracy improved (by 0.003% for Siamese$_{L1}$ and 0.002% for Siamese$_{\oplus}$), but also the false negative rate of our DL-based model has significantly decreased (by 38.462% for Siamese$_{L1}$ and 53.846% for Siamese$_{\oplus}$). According to the results, Siamese$_{\oplus}$ displays superior performance with the Stanislaus dataset, while Siamese$_{L1}$ performs better on the Merced dataset. This is due to Siamese$_{\oplus}$ is particularly effective when the training data is sufficient, as it learns

how to fuse features instead of manual engineering. On the other hand, Siamese$_{LI}$ is better equipped to handle situations where the training data is insufficient, as it is guided to focus specifically on discrepancies. As demonstrated in Fig. 11, when enhanced with the proposed Siamese architecture, our DL-based mark segmentation model is successful in detecting the marks, which are underrepresented in the training dataset. Without the use of the Siamese network architecture, the DL mark segmentation model fails to detect these marks.

TABLE III: The result of Siamese-based DL

Test Dataset		Siamese$_{LI}$			Siamese$_{\oplus}$		
Name	# of Marks	FP	FN	ACC	FP	FN	ACC
Stanislaus	2,820	1	2	99.894%	2	0	99.929%
Merced	89,960	3	6	99.990%	5	6	99.988%
Overall	92,780	4	8	99.987%	7	6	99.986%

VI. Conclusions and Future Works

This paper proposes the first highly automated ballot tabulation methods, including a traditional computer vision and a deep learning-based methods. In the T-CV mark segmentation model-based method, we further propose the first highly automated ballot image registration algorithm and voting target localization algorithm. We also propose a Siamese network architecture to improve the DL-based mark segmentation model's capacity of handling the marks underrepresented in training dataset.

In the future, we see at least three ways to further improve the DL-based mark detection and segmentation method: First, we intend to design transformer-based network architectures to replace the current convolutional neural network-based (CNNs) backbone. Unlike CNNs, which process data with local receptive fields, transformers with attention mechanism allow unrestricted interactions between each patch and every other patch in the image. This could be particularly beneficial for mark detection and segmentation tasks where the model needs to consider global context and interdependencies among different regions of the ballot image. Secondly, we plan to explore other model architectures such as U-Net or YOLO (You Only Look Once). U-Net, with its encoder-decoder structure, has been shown to perform well in tasks that require precise segmentation. On the other hand, YOLO, an architecture designed for real-time object detection, could potentially improve the efficiency of our mark detection and segmentation tasks. Another direction of future work is designing a diffusion-based image registration model to address the misalignment issue that arises between a hand-marked ballot and its corresponding blank template. Diffusion models, a kind of generative model, are effective in modeling the distribution of data and generate new data instances. They could potentially be used to generate well-aligned blank template ballots corresponding to specific hand-marked ones. This approach might significantly improve the speed and efficacy of our image registration process.

References

[1] K. Wang, N. Carlini, E. Kim, I. Motyashov, D. Nguyen, and D. A. Wagner, "Operator-assisted tabulation of optical scan ballots." in *EVT/WOTE*, 2012.

[2] A. Bajcsy, Y.-S. Li-Baboud, M. Brady *et al.*, "Systematic measurement of marginal mark types on voting ballots," Technical report, National Institute for Standards and Technology, Tech. Rep., 2015.

[3] E. Kim, N. Carlini, A. Chang, G. Yiu, K. Wang, and D. Wagner, "Improved support for machine-assisted ballot-level audits," in *2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 13)*, 2013.

[4] E. H. B. Smith, D. Lopresti, and G. Nagy, "Ballot mark detection," in *2008 19th International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–4.

[5] P. Xiu, D. Lopresti, H. Baird, G. Nagy, and E. B. Smith, "Style-based ballot mark recognition," in *2009 10th International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 216–220.

[6] C.-L. Liu, H. Sako, and H. Fujisawa, "Discriminative learning quadratic discriminant function for handwriting recognition," *IEEE Transactions on Neural Networks*, vol. 15, no. 2, pp. 430–444, 2004.

[7] S. Barretto, W. Chown, D. Meyer, A. Soni, A. Tata, and J. A. Halderman, "Improving the accuracy of ballot scanners using supervised learning," in *International Joint Conference on Electronic Voting*. Springer, 2021, pp. 17–32.

[8] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," in *Science and Information Conference*. Springer, 2019, pp. 128–144.

[9] A. Cordero, T. Ji, A. Tsai, K. Mowery, and D. A. Wagner, "Efficient user-guided ballot image verification." in *EVT/WOTE*, 2010.

[10] S. Divya, S. Paul, and U. C. Pati, "Structure tensor-based sift algorithm for sar image registration," *IET Image Processing*, vol. 14, no. 5, pp. 929–938, 2019.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[12] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Joint pattern recognition symposium*. Springer, 2007, pp. 214–223.

[13] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "An improved algorithm for tv-l 1 optical flow," in *Statistical and geometrical approaches to visual motion analysis*. Springer, 2009, pp. 23–45.

[14] M. A. Mohamed and B. Mertsching, "Tv-l1 optical flow estimation with image details recovering based on modified census transform," in *International Symposium on Visual Computing*. Springer, 2012, pp. 482–491.

[15] J. Matas, C. Galambos, and J. Kittler, "Progressive probabilistic hough transform," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 1998, pp. 26.1–26.10, doi:10.5244/C.12.26.

[16] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[17] F. Bolelli, S. Allegretti, and C. Grana, "One dag to rule them all," 2021.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[19] Joblib Development Team, "Joblib: running python functions as pipeline jobs," 2020. [Online]. Available: https://joblib.readthedocs.io/