2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR) | 979-8-3503-5142-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/MIPR62202.2024.00021

GenCheck: A LoRA-Adapted Multimodal Large Language Model for Check Analysis

Fei Zhao[†*], Jiawen Chen[‡*], Bin Huang[§], Chengcui Zhang[†], Gary Warner[†],
Rushi Chen[¶], Shaorou Tang[‡], Yuanfei Ma[‡], Zixi Nan[‖]
[†] The University of Alabama at Birmingham, Birmingham, USA
[‡]Beijing-Dublin International College, Beijing University of Technology, Beijing, China
[§] Beijing Hua Yu Xin Zhang Technologies Co., Ltd, Beijing, China
[¶]College of Information Science and Technology, Shihezi University, Shihezi, China
[‖]Computer Science and Technology College, University of Chinese Academy of Sciences, Beijing, China
larry5@uab.edu, cnjywr@emails.bjut.edu.cn, huangbin.cn@gmail.com, {czhang02, gar}@uab.edu,
20231008237@stu.shzu.edu.cn, {tangshaorou, 22550428}@emails.bjut.edu.cn, nanzixi23@mails.ucas.ac.cn
*These authors contributed equally to this work.

***Abstract*—Rising incidences of paper check fraud, particularly with checks illicitly sold on platforms such as Telegram, pose significant challenges in financial security. Despite investigators' capability to gain access to these platforms, manually pinpointing checks in images and extracting necessary details to alert banks are inefficient and unscalable. Traditional optical character recognition-based (OCR) systems for extracting textual details from checks specifically struggle with handwritten content and are constrained by their dependency on predefined check layouts, limiting their effectiveness across varied and evolving check designs. To address these challenges, we introduce GenCheck, a generative AI-based framework that automates both the check detection and accurate extraction of check information, ensuring robust performance across various check layouts or styles. GenCheck operates through a two-stage pipeline: the preliminary stage encompasses multiple sub-tasks including check image classification, single check segmentation, image rectification, and check element detection, while the main stage focuses on the key task of check information extraction. Central to our pipeline is the strategic enhancement of a state-of-the-art (SOTA) multimodal large language model (LLaVA-NeXT) using Low-Rank Adaptation (LoRA). This fine-tuning leverages the model's pre-trained knowledge, applying a targeted, parameter-efficient approach that significantly enhances its ability to accurately extract key details such as dates, amounts, and payee information from paper check images. Our framework achieves exceptional accuracy rates in extracting date information with 92.07% for year, 85.16% for month, and 82.72% for day. It also obtains an accuracy of 80.61% in extracting monetary amounts and a normalized edit distance of 0.2583 for payee information, demonstrating substantial improvements over pure OCR-based methods. As the first framework of its kind, GenCheck establishes a methodological base that supports continuous innovation and enhancement, allowing for independent updates of each component model. This also sets a new standard in automated check analysis, reducing the need for labor-intensive, rule-based processes and significantly advancing fraud prevention initiatives.**

***Index Terms*—Multimodal Large Language Model, Handwritten Check, Deep Learning, Low-Rank Adaptation, Check Fraud**

I. Introduction

Paper checks, as a common medium in financial transactions, are vulnerable to a range of fraudulent activities, primarily due to their physical and often lightly regulated nature. In recent years, platforms such as Telegram have become hotspots for criminal activities where stolen checks are frequently sold. Investigators currently rely on manual methods to scan Telegram channels, identify images containing checks, and extract relevant information such as account details and amounts to alert banks or other institutions about potential fraud. However, this manual inspection process is not only labor-intensive but also fails to scale with the increasing volume of check fraud. The primary challenges in automating this process include the lack of an automated framework specifically designed for the detection and extraction of information from paper checks often handwritten, which often feature diverse and unique layouts. Additionally, traditional optical character recognition-based (OCR) systems, while useful, are hindered by their inability to handle the variability of handwritten content effectively and their reliance on predefined check layouts to categorize extracted content into specific classes such as payee, payer, and date.

Recognizing these gaps, we propose GenCheck, a groundbreaking framework that leverages generative AI to automate the detection and information extraction from paper check images. GenCheck transcends traditional OCR limitations by employing a visual question answering (VQA) approach with the LoRA-adapted LLaVA-NeXT model. This adaptation allows us to utilize the rich, pre-trained knowledge of the multimodal large language model (MLLM) and apply targeted, efficient parameter tuning that significantly enhances the model's capability to process visual and textual content from checks. Meanwhile, LoRA offers a storage-efficient approach for handling multiple information extraction tasks. It allows reusing a single, pre-trained MLLM while storing task-specific LoRA weights. This eliminates the need to save separate

2770-4319/24/$31.00 ©2024 IEEE
DOI 10.1109/MIPR62202.2024.00021

MLLMs for each target element (e.g., date, money amount, and payee).

Our contribution can be summarized as follows:

1. GenCheck: This is the first automated and generative AI-based framework dedicated to check detection and information extraction. It is also the pioneer in applying visual question answering (VQA) techniques to this domain, in which its textual prompt enables the extraction of critical details, e.g., payee and money amount, from checks without the need for hardcoded formatting rules. Compared to pure OCR-based systems, our LoRA-adapted MLLM demonstrates substantially improved performance, particularly in accurately extracting content that can be written in diverse formats, e.g., date. This not only showcases the superiority of GenCheck over OCR but also sets a new benchmark for the integration of AI in financial security operations.

2. A Two-Stage Pipeline: Our solution introduces a two-stage pipeline that significantly enhances processing. The preliminary stage includes sub-tasks/steps as check image classification, single check segmentation, image rectification, and check element detection. The second and the main stage focuses on essential check information extraction. The preliminary stage efficiently isolates areas of interest by removing irrelevant content, such as photo backgrounds, allowing the framework to focus on task-specific regions. As the first framework of its kind, GenCheck establishes a methodological base that supports continuous innovation and enhancement, allowing for independent updates of each component model.

3. Check Image Benchmark Dataset: We have developed the first comprehensive dataset tailored specifically for paper check images, featuring 8,689 images from over 15 financial institutions. This dataset is **organized into sub-datasets** for five distinct tasks: check image classification, single check segmentation, image rectification, check element detection, and information extraction. It includes annotations for 11 key elements such as payee name, date, amount, and drawer name, complete with bounding box data for each element on each image. This dataset underpins the development of our modular models, each designed for independent upgrade to boost overall system efficacy.

The remainder of this paper is organized as follows: Section II introduces background and reviews existing related methodologies. Section III details the methodology of our two-stage GenCheck pipeline. Section IV introduces our dataset collection. Section V presents the experimental results, demonstrating GenCheck's effectiveness against traditional methods, and discusses the implications of our findings. Section VI summarizes contributions and provides future research directions.

II. Background and Related Work

A. Paper Check Information Extraction

Effective information extraction from paper check images is critical for check fraud prevention. Traditional systems often focus narrowly on specific aspects of check processing and fall short in comprehensive security measures. For example, systems such as the one analyzed in [1] detect fraudulent patterns based on transaction amounts and **overlook other critical details**, failing when fraudsters mimic legitimate patterns with altered payee information. Similarly, the Anti-fraud Tensorlink4cheque model (AFTL4C) [2], which uses generative adversarial networks to distinguish between real and fake checks, struggles with images of genuine checks physically modified by fraudsters as it can only detect synthetic images. Meanwhile, it is not capable of extracting detailed information from check images. Additionally, methods such as those in [3] focus on segmenting numeric data from checks using deep neural networks but do not address non-numeric data extraction or context-specific categorization such as dates or routing numbers. The A2iA Check Reader system [4] also faces similar limitations, effectively categorizing numeric segments but unable to process letters or to extract comprehensive check information. The above underscors the urgent need for an advanced, multifaceted approach to check information extraction.

B. Object Detection and Segmentation

The effectiveness of object detection and segmentation models is pivotal in applications requiring precise element isolation, such as scanned check processing. The authors in [5] propose Mask R-CNN that offers high accuracy in instance segmentation by extending Faster R-CNN [6] to predict segmentation masks for each Region of Interest (RoI). However, its high computational demand and long inference time limit its practicality for real-time applications. Similarly, the authors in [7] propose Segment Anything Model which adaptively handles a wide range of objects and scenarios without prior specification of the target categories. While this model demonstrates remarkable flexibility in segmentation tasks, it also demands high computational resources and exhibits substantial inference time, which can hinder its deployment [8]. The authors in [9] propose UNet mdoel, which is primarily designed for medical image segmentation, noted for its simplicity and speed, but is typically unsuited for instance segmentation tasks that require differentiation between individual objects of the same class. YOLO (You Only Look Once) [10], a SOTA lightweight object detection and instance segmentation model, has a one-stage design striking an optimal balance between speed and accuracy.

C. Multimodal Large Language Models

The integration of vision backbones with language models in large multimodal foundation models has significantly advanced their capabilities in vision-language tasks. Notably, models such as Flamingo [11] and BLIP-2 [12] have demonstrated exceptional capabilities in general multimodal tasks. However, their performance in OCR tasks remains limited due to a lack of OCR-specific training data. In contrast, ChatGPT-4 [13], while capable of interpreting handwritten content from images, presents concerns regarding data privacy due to its online deployment. To address these limitations, the LLaVA-NeXT model [14] has been specifically designed to excel in handwritten OCR tasks. It leverages a training dataset enriched

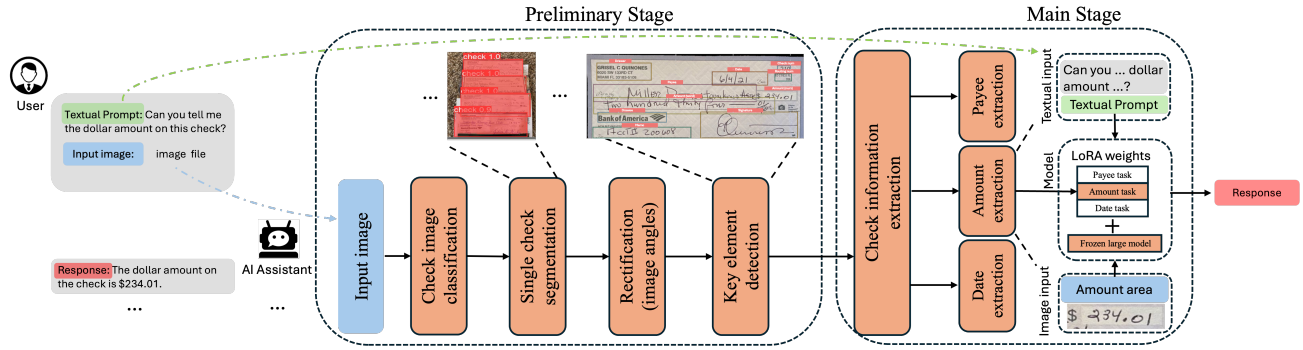

Fig. 1. Visualization of our two-stage pipeline for paper check analysis using the LoRA-adapted MLLM. In the preliminary stage, the input image is processed to remove irrelevant content and focus on areas of interest, preventing the model from being distracted by extraneous details. This refined image, along with the textual prompt, is then sent to the LoRA-adapted MLLM in the main stage to generate accurate responses. This approach mitigates the risk of incorrect predictions that often occur when using unprocessed, whole images as inputs.

with OCR samples, ensuring both improved OCR accuracy and enhanced data privacy measures, thus filling a critical gap in current multimodal model applications. To maximize MLLM's efficiency and adaptability for specific tasks, Low-Rank Adaptation (LoRA) is proposed by the authors in [15]. This parameter-efficient fine-tuning (PEFT) technique focuses on refining critical parameters, optimizing LLaVA-NeXT's performance while conserving computational resources. This strategic enhancement ensures that GenCheck benefits from both the advanced capabilities of LLaVA-NeXT and the efficiency of LoRA, facilitating a scalable and effective solution for real-world check processing challenges.

Given the limitations of existing methods, we propose GenCheck, a two-stage framework that harnesses the precision of object detection model (YOLO) and the data processing prowess of MLLM (LLaVA-NeXT), fine-tuned with Low-Rank Adaptation (LoRA). This framework excels at identifying and segmenting check images, removing irrelevant details, and extracting information accurately without the constraints of predefined layouts.

III. Methodology

Our framework for automated check information extraction is detailed in Fig. 1. GenCheck is structured into two main stages to ensure efficient processing of images collected from check fraud channels, which often have low-to-medium resolutions, include multiple checks laid out in different orientations within a single image, and often have missing fields intentionally removed by fraudsters, hindering the downstream information extraction.

A. Preliminary Stage

Four sequential steps are in this stage guiding the framework to focus on areas of interest:

1) Check Image Classification Model: The first step in our pipeline involves classifying images into three categories: no-check, single-check, and multiple-checks using a customized YOLO_cls [10] model. This classification aids in filtering out images without checks, thus focusing the subsequent processing steps only on relevant images. Its effectiveness is evaluated in Section V-B1.

2) Single Check Segmentation Model: For images classified as containing checks, the next step involves segmenting individual check areas from the image. This task is crucial when an image contains multiple checks, and it ensures that each check is processed individually, minimizing interference from adjacent checks. We utilize a customized YOLO_seg model to recognize and isolate these areas accurately. Examples are shown in Fig. 1.

3) Image Rectification Method: Since all the images in the dataset are collected from Telegram's channel messages, variations in camera angle and in check segment layout orientations often lead to noticeable skewness and perspective distortions. To correct these, we first deploy a customized YOLO_rec model that is designed to detect 72 distinct classes of rotation degrees (each class covering a 5-degree interval across the full 360-degree spectrum). Following the YOLO_rec coarse-level correction, we refine the skewness correction by utilizing Line Segment Detector (LSD) to identify prominent lines within the image (longer than one-fifth of the maximum of the check's width or height) to determine the predominant inclination angle guiding the fine-level rotation adjustment to the check, ensuring optimal alignment for subsequent processing steps. This coarse-to-fine two-step rectification process effectively overcomes the limitation of LSD algorithm that is mostly only effective in the rotation range of [-90, 90] degrees, attributed to the maximum error margin of 14 degrees of the coarse-level correction, and is therefore able to handle excessive rotation of check segments in images.

4) Check Element Detection Model: Before extracting information, it is vital to locate 11 key elements on the checks, including date, payer, and payee fields (see Fig. 3), etc. This step uses another customized YOLO_det model trained to detect these elements' locations (bounding boxes), which also helps in overcoming the limitations imposed by predefined check layouts. This model automatically detects and outputs bounding boxes for the regions of interest, preparing them for

information extraction down the road. Examples are shown in Fig. 1.

B. Main Stage

1) Check Information Extraction with PEFT of MLLM: The final stage of our pipeline involves extracting check information via a visual question answering (VQA) task, where sub-images defined by previously detected bounding boxes are processed as shown in Fig. 1. These images, alongside contextual textual prompts, are fed into our LoRA-adapted LLaVA-NeXT model, which has been fine-tuned specifically for this task. Each element, such as payee, date, and amount, benefits from specialized LoRA weights that are fine-tuned on datasets tailored to each field (e.g., payee VQA set, amount VQA set, and date VQA set). This configuration allows us to extract intricate details from each check element dynamically, enhancing the model's performance on each task. Since the frozen LLaVA-Next model can be reused for all elements, only lightweight LoRA weights for each element need to be stored. In our framework, LoRA enhancements are strategically applied only to the Mistral component of LLaVA-NeXT, which has 7.3 billion parameters and focuses on language processing. We avoid applying LoRA modifications on CLIP (vision tower of LLaVA-NeXT) to maintain the integrity of its visual processing capabilities. This targeted application of LoRA enhances the model's textual comprehension and contextualization without necessitating additional training for vision-language alignment. LoRA's application across all linear layers of the Mistral model allows for precise, targeted improvements in processing check-related data as shown in Fig. 2. The hyperparameters for LoRA, i.e., LoRA's rank and scaling factor, are set to 128 and 256, respectively, optimizing the balance between model adaptability and computational efficiency. These settings fine-tune the model's ability to interpret and utilize pre-trained knowledge specifically for financial transactions, ensuring high accuracy and context relevance in outputs.

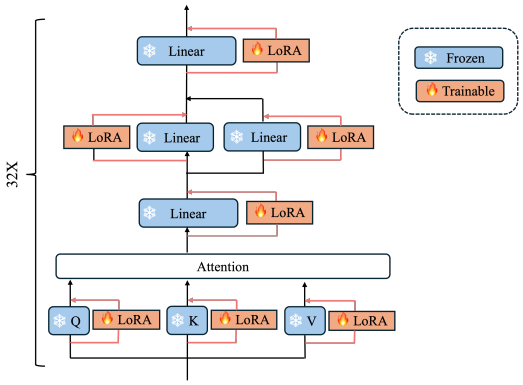

Fig. 2. Schematic of LoRA layers: this figure depicts LoRA layers integrated alongside linear layers. Normalization layers and skip-connections are omitted for clarity.

2) VQA-based Information Extraction Strategy: The adoption of a visual question answering (VQA) strategy in our framework addresses the challenges posed by the diverse layouts and formats of content (especially the handwritten ones) on checks. Traditional OCR methods struggle with varying formats used to represent fields such as dates, which may appear with different separators ("-", "/", ",") and styles ("5-20-24", "05/20/2024", "May 20, 2024"). These require complex hardcoded rules for accurate interpretation. Using a VQA approach with our multimodal large language model (MLLM), we utilize textual prompts such as "What is the date in the format Month: MM, Day: DD, Year: YYYY?" to guide the model's output formatting. This strategy allows the MLLM to utilize its pre-trained knowledge to recognize and standardize diverse date notations efficiently. Unlike rigid OCR systems, our model dynamically adapts to various handwriting styles and data representation formats, enhancing flexibility and robustness. This adaptability, facilitated by prompt engineering, ensures our system's effectiveness across non-standardized check formats and writing patterns, offering a robust solution to real-world challenges.

IV. Datasets

With the support of Dark Tower company, we curated a highly diversified dataset of 8,689 images sourced from various Telegram channels. This dataset exhibits challenging scenarios, such as overlapping checks in one image, which significantly complicates the detection and information extraction tasks. We meticulously annotated these images for 11 information elements, including bounding boxes and content, as illustrated in Fig. 3. To rigorously evaluate both our entire framework and its individual components, we structured the data into five distinct sub-datasets throughout the collection and model development phases.

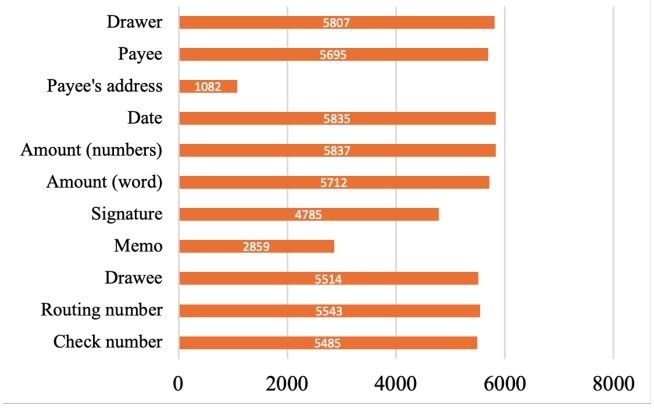

Fig. 3. The instance count of each element in our dataset. Not every field is visible in every check due to object occlusion and/or intentional removal of certain fields by fraudsters.

1) Image Categorization Dataset: The primary categorization divides the input images into three classes based on their content: Multiple Checks (images containing more than one

check), Single Check (images containing exactly one check), and No Checks (images not containing any checks). This categorization ensures that images without checks are filtered out early in the pipeline, thereby optimizing the processing resources. We curated a dataset comprising 489 images with multiple checks, 6,818 images with a single check, and 1,382 images without any checks. The training to testing data ratio is approximately 10:1 for each class.

2) Single Check Segmentation Dataset: Single check segmentation process is vital for isolating each check in images that contain multiple check segments to prevent data crossover during the extraction phase. With the assistance of the Segment Anything Model, we collected ground truth data for single check segmentation across a total of 456 check images, each containing multiple checks. The dataset was divided into 1,468 training samples from 361 check images and 362 testing samples from 95 check images.

3) Check Image Rectification Dataset: We constructed a dataset for the check image rectification experiment, consisting of 92,160 rotated variations from 256 original images perfectly rectified. Each image was rotated from 0 to 359 degrees, incrementing the rotation by one degree per iteration. The resulting images were then categorized into 72 classes based on a 5-degree interval. This dataset will be utilized to evaluate the performance of our check image rectification methods in Section V-B3.

4) Check Element Detection Dataset: To facilitate accurate extraction of information from a check, it is essential to identify specific element areas such as date, payee, and amount. We compiled bounding box ground truth data for 11 key elements across 6,235 images for training, 504 for validation, and 504 for testing. This dataset is used to tune SOTA models to accurately locate those elements within each check segment.

5) Information Extraction Dataset: The final dataset focuses on the recognition of text within the annotated elements of checks. We have enriched the Check Element Detection Dataset by including text content ground truth for each element within every check. This is essential for training our MLLM models to accurately extract textual content from identified elements. The enriched dataset comprises 6,235 images for training, 504 for validation, and 504 for testing.

V. Results and Discussions

A. Main Stage Evaluation

This section presents the results and discusses the efficacy of our GenCheck framework in the extraction of check information. We compare the performance of our proposed LoRA-adapted MLLM, Zero-shot MLLM, and a contemporary OCR-based method (PaddleOCR [16]). Three fields, i.e., date, money amount, and payee name, are selected as representatives of date, number, and string typed fields among the 11, and used in the evaluation experiments. Another reason those three are selected is that a large portion of checks in our dataset have those three fields hand written, a challenging test bed for any OCR system.

1) Date Element Extraction: We evaluated the accuracy for extracting the year, month, and day components from the checks in the Information Extraction Dataset, where accuracy is measured based on a strike, meaning a 100% exact match is required. The results are summarized in Table I. These results distinctly highlight the superior performance of our LoRA-adapted MLLM. Notably, the model achieves over 85% accuracy in extracting the year and month components from handwritten checks, significantly outperforming the SOTA optical character recognition-based (OCR) method and the Zero-shot MLLM. Specifically, the LoRA-adapted MLLM exhibits substantial improvements over the OCR-based method, with accuracy increase of approximately 174.50% for the year, 86.22% for the month, and 92.87% for the day. Compared to the Zero-shot MLLM, our model shows improvements of about 74.91% for the year, 34.73% for the month, and 50.18% for the day.

TABLE I
Date Element Extraction: Accuracy

Model	Accuracy ↑		
	Year	Month	Day
OCR-based method	33.54%	45.73%	42.89%
Zero-shot MLLM	52.64%	63.21%	55.08%
LoRA-adapted MLLM	**92.07%**	**85.16%**	**82.72%**

This substantial enhancement underscores the effectiveness of parameter-efficient fine-tuning (PEFT) applied to MLLMs. The lower accuracy rate for the day, relative to the year and month, can be caused by the positioning of the day within the check's format. The day component often appears between two surrounding "/" separators, commonly used in data formatting, which closely resemble the handwritten numeral "1". For example, given the handwritten date: "5/2/2023", the system could get confused and output a wrong result on day as either "12" or "21".

2) Payee Element Extraction: The normalized edit distance (NED) measure is used for evaluating the accuracy of extracted payee information, with results summarized in Table II. Our LoRA-adapted MLLM demonstrates a marked improvement in NED, achieving a value of 0.2583. This performance substantially surpasses that of the SOTA optical character recognition-based (OCR) method and the Zero-shot model, with reductions in NED by 0.2138 and 0.1498, respectively.

TABLE II
Payee Element Extraction: Normalized Edit Distance

Model	Normalized Edit Distance ↓
OCR-based method	0.4721
Zero-shot MLLM	0.4081
LoRA-adapted MLLM	**0.2583**

These results illustrate the superior capability of our LoRA LLM in handling the complexities of payee information often handwritten, where contextual knowledge can significantly influence recognition accuracy. Despite challenges in manual annotation, where ambiguity due to similar-looking letters

and incomplete contextual knowledge may lead to imperfect ground truth data, our model still exhibits reasonable performance on the current ground truth. Worth mentioning, in some instances, the predictions by our framework have proven more accurate than the ground truth annotations themselves, indicating not only the effectiveness of our model but also its potential to correct or identify errors in manual annotations.

3) Amount Element Extraction: The evaluation of amount extraction from checks relies on strike-based accuracy, introduced in Section V-A1, and normalized mean average error (NMAE), with outcomes presented in Table III. Our LoRA-adapted MLLM achieves a substantial lead in both metrics, with an normalized mean average error of 0.1027 and an accuracy of 80.61%. This represents a significant improvement over both the OCR-based method and the Zero-shot MLLM. Specifically, our model reduces the NMAE by 0.2999 compared to the OCR-based method and by 0.2404 compared to the Zero-shot MLLM, while also enhancing accuracy by 64.21% and 83.87%, respectively.

TABLE III
AMOUNT ELEMENT EXTRACTION: NMAE AND ACCURACY

Model	NMAE ↓	Accuracy ↑
OCR-based method	0.4026	49.09%
Zero-shot MLLM	0.3431	43.84%
LoRA MLLM	**0.1027**	**80.61%**

The comparative analysis also reveals that the Zero-shot MLLM, despite having a lower accuracy of 43.84% compared to the OCR-based method's 49.09%, achieves a better NMAE score. This mismatch suggests that while the Zero-shot model's predictions are closer to the ground truth values, they less frequently match the ground truth exactly compared to those of the OCR method. However, after applying parameter-efficient fine-tuning (PEFT) to the MLLM for the task of amount information extraction, our LoRA-adapted MLLM successfully addresses this precision gap.

B. Preliminary Stage Evaluation

1) Check Image Classification: As shown in Table IV, the model achieved at least an F1 score of 97.96% across these classes on the Image Categorization Dataset, indicating excellent precision and recall.

TABLE IV
IMAGE CLASSIFICATION RESULTS USING YOLO_CLS MODEL

Performance	Multiple	Single	No Checks
Precision ↑	97.96%	99.56%	100%
Recall ↑	97.96%	99.85%	98.56%
F1 Score ↑	97.96%	99.70%	99.27%

2) Single Check Segmentation: In this experiment, we tested our YOLO_seg model on the Single Check Segmentation Dataset. Our model achieves an average precision (AP) [10] score of 92.5% at an IoU threshold of 0.8, demonstrating high effectiveness in segmenting checks accurately.

3) Check Image Rectification: We evaluate our rectification method on the Check Image Rectification dataset. The accuracy of this course-to-fine rectification method achieves 88.10%. As shown in Fig. 4 the distribution of the angle error degree between the predicted and the ground truth output by the course-level correction, the average error is 3.5 degrees and the maximum error angle is only 14 degrees. By rotating the image according to the YOLO_rec classification model's output, we ensure the image's inclination angle is contained within -90 to 90 degrees, effectively resolving vertical inversion issues of LSD.

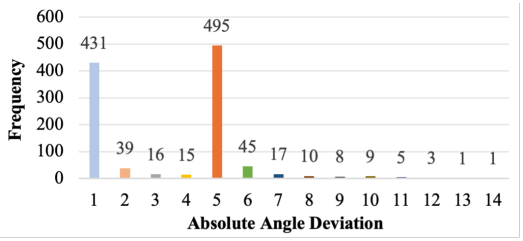

Fig. 4. Histogram of absolute angular deviations

4) Check Element Detection: Our YOLO_det model's detection performance was evaluated by the AP and mean average precision (mAP) [10] at Intersection over Union (IoU) thresholds 0.7 on the Check Element Detection Dataset. As shown in Table V, our model obtains an mAP score of 92.2%.

5) Discussion: Each preliminary task's performance underscores the efficacy of the specialized models trained for specific components in check processing. High accuracy rates in classification and segmentation ensure that only relevant check images and areas are forwarded for detailed extraction, minimizing potential errors from irrelevant or misaligned data. The precise detection of key check elements further facilitates accurate and efficient information extraction in the final stage, demonstrating the holistic strength of the GenCheck framework.

Unlike date, amount, and payee fields, the drawer, drawee, and check number fields are commonly and largely printed. Therefore, we test information extraction from these elements using traditional OCR-based methods. Notably, the drawee element presents a unique challenge. We have collected over 500 names of financial institutions from our annotated dataset and attempted to match each check's issuing institution based on the edit distance between the collected institution names and the text extracted by the OCR-based method. The results for drawer, drawee, and check number were reasonably accurate, as shown in Table VI. Given the satisfactory performance of traditional OCR-based methods and the limited potential for further enhancements compared to elements such as payee, amount, and date, further applying our MLLM-based framework to these elements is deemed unnecessary at this stage.

VI. CONCLUSIONS AND FUTURE WORK

This study has presented GenCheck, an innovative, generative AI-based framework for automated detection and extrac-

TABLE V
CHECK ELEMENT DETECTION RESULTS USING YOLO_DET

Model	Drawer	Payee	Payee's Address	Date	Amount	Signature	Memo	Routing Number	Check Number	mAP
Yolov8n	94.6%	92.2%	94.8%	96.9%	95.0%	93.3%	78.4%	93.7%	96.3%	92.2%

TABLE VI
OCR-BASED EXPERIMENT RESULTS

Model	Normalized Edit Distance ↓		Accuracy ↑
	Drawer	Check Number	Drawee
OCR-based method	0.1715	0.1197	80.17%

tion of information from paper check images with varying formats, layouts, and resolutions collected from various check fraud telegram channels. By leveraging the SOTA LLaVA-NeXT model with LoRA, our approach significantly outperforms the OCR-based method and sets new benchmarks in accuracy and reliability for processing financial documents. The framework effectively addresses the complexities of extracting check elements such as date, amount, and payee details, demonstrating a robust capability to adapt to diverse check layouts and handwriting styles. Our results underscore the effectiveness of our two-stage multi-step pipeline, particularly in enhancing the precision and scalability of check processing tasks.

Looking forward, we aim to (1) extend the parameter-efficient fine-tuning on MLLM to additional check elements such as Drawer and Drawee, enhancing the comprehensiveness and accuracy of our check information extraction, (2) to adapt the LoRA-enhanced MLLM to process entire check images rather than cropped subimages, aiming to reduce errors from preliminary detection steps and to exploit the non-local relationship among fields/elements on a check image, and (3) to build a multi-agent-based framework, with each agent specialized for a distinct step of the processing pipeline, improving each step's robustness and accuracy.

VII. ACKNOWLEDGEMENT

The authors would like to thank Ashlynn Schultz from DarkTower for sharing this problem and the raw check image dataset with our team.

This work was supported in part 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] K. Julisch, "Risk-based payment fraud detection risk-based payment fraud detection," tech. rep., IBM Research, 2010.

[2] P. Uyyala and D. C. Yadav, "The advanced proprietary ai/ml solution as anti-fraudtensorlink4cheque (aftl4c) for cheque fraud detection," *The International journal of analytical and experimental modal analysis*, vol. 15, no. 4, pp. 1914–1921, 2023.

[3] R. Palacios, A. Gupta, and P. S. Wang, "Handwritten bank check recognition of courtesy amounts," *International journal of image and graphics*, vol. 4, no. 02, pp. 203–222, 2004.

[4] N. Gorski, V. Anisimov, E. Augustin, O. Baret, D. Price, and J.-C. Simon, "A2ia check reader: A family of bank check recognition systems," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, pp. 523–526, IEEE, 1999.

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

[6] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[7] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[8] S. Xu, H. Yuan, Q. Shi, L. Qi, J. Wang, Y. Yang, Y. Li, K. Chen, Y. Tong, B. Ghanem, *et al.*, "Rap-sam: Towards real-time all-purpose segment anything," *arXiv preprint arXiv:2401.10228*, 2024.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241, Springer, 2015.

[10] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO (version 8.0.0) [computer software]." https://github.com/ultralytics/ultralytics, 2023.

[11] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in neural information processing systems*, vol. 35, pp. 23716–23736, 2022.

[12] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*, pp. 19730–19742, PMLR, 2023.

[13] OpenAI, "Chatgpt-4." https://www.openai.com/, 2023. Accessed: 2023-12-01.

[14] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, "Llava-next: Improved reasoning, ocr, and world knowledge," January 2024.

[15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[16] PaddlePaddle, "Optical character recognition toolkits." https://github.com/PaddlePaddle/PaddleOCR/tree/main, 2024. Accessed: 2024-03-15.