FieldChartOCR: Extraction of Handwritten Charts and Tables

Project Proposal by: Chris Brechin

Prepared for: DTSA-5506

Date: November 12th, 2025

Abstract

Handwritten charts and technical notes are rich with quantitative insight yet remain locked inside scanned pages. FieldChartOCR extends ChartOCR to converting hand-drawn charts and tables into structured datasets. The procedure goes through chart classification, element detection, and handwriting OCR, to information extraction. The focus is on delivering a practical tool that field technicians and researchers can rely on without exhaustive manual transcription.

1. Introduction

Digitizing handwritten technical notebooks is labour-intensive. Engineers, students, and analysts rely on annotations, sketches, and mixed layouts that resist automated extraction. Existing OCR tools, like ChartOCR, provide plain text at best; they ignore hand-drawn informatics in order to optimize accuracy. On the other hand, professionals (such as field technicians) must manually re-create charts in tools such as Excel, Desmos, or CAD systems—a significant barrier to iteration and collaboration.

The goal of FieldChartOCR is to build a unified pipeline that ingests handwritten graphs, tables, and mechanical annotations and produces structured artifacts suitable for analytics and visualization. By coupling deep learning with rule-based reasoning, I hope to generalize across chart families while maintaining transparency in intermediate outputs. Delivering this capability advances document understanding, accelerates knowledge sharing, and supports accessibility by providing machine-readable alternatives to complex figures.

2. Related Work

- **Chart Understanding Frameworks:** ChartOCR's hybrid keypoint-and-rule architecture demonstrates the effectiveness of combining deep detection with domain logic. ChartSense and ReVision apply rule-based heuristics but struggle with diverse layouts. DVQA and FigureQA focus on question answering but assume neatly rendered charts.
- **Diagram & Table Recognition:** DiagramParseNet and other graph-parsing models detect nodes and edges in structured diagrams. DeepDeSRT and PubTabNet handle printed tables yet rely on sharp lines and consistent fonts, unlike the irregular strokes in hand-drawn notes.
- **Handwriting OCR & Math Recognition:** Transformer-based handwriting recognizers and MathOCR systems convert cursive and symbolic notation into LaTeX. They typically operate on grayscale inputs and overlook complex layouts.
- **Document Normalization & Imaging:** Research on illumination correction and phase-based representations inspires our preprocessing layer. Multispectral imaging studies inform our roadmap for future sensing modalities.

3. Proposed Work

3.1 Data Sources

- handwritten notebooks and diagrams.
- corresponding digital diagrams and comma separated list of data for training validation
- In order to avoid informational leaks, I will synthesize my datasets to cover varied topics of study.

3.2 System Architecture

1. **Preprocessing & Layout Normalization**
   - Perform denoising, skew correction, grid suppression, and edge enhancement tailored to notebook paper.
   - Standardize contrast and dynamic range to support downstream detectors.
   - **Note:** I may move to scanning the images if this part proves too onerous.
2. **Chart Type Classification**
   - Lightweight CNN-transformer hybrid predicts dominant chart categories for each region.
     - The goal is to match the 3 charts recognized by ChartOCR: bar charts, line charts, and pie charts.
   - Supports mixed-content pages by assigning probabilities per region.
3. **Element Identification & OCR**
   - Shared detection backbone with heads for axes, scales, titles, data marks, and table structure.
4. **Semantic Assembly & Output**
   - Associates text blocks with detected elements, enforces basic geometric constraints, and exports Markdown summaries plus CSV datasets.

3.3 Development Plan

- **Phase 1:** Build preprocessing pipeline, layout normalization, and table-to-CSV baseline.
- **Phase 2:** Train chart classifier and core element detectors; integrate handwriting OCR.

- **Phase 3:** Finalize semantic assembly, polish outputs, and iterate on error analysis.

4. Evaluation Plan

- **Effectiveness Metrics**
    - Chart classification macro-F1 across core categories (bar, line, scatter, tables).
    - Element detection precision/recall for axes, titles, and data marks.
    - OCR character error rate (CER) and table reconstruction F1.
- **Efficiency Metrics**
    - End-to-end runtime per page and memory footprint.
- **Experimental Setup**
    - Stratified train/validation/test split by chart type and handwriting style.
    - Baselines: ChartOCR, generic OCR plus heuristic extraction, and manual transcription samples.
    - Ablations on preprocessing and semantic assembly.

5. Timeline & Milestones

Week	Milestone
1	Finalize annotation schema, label initial dataset, implement normalization prototype.
2	Integrate baseline handwriting OCR and chart classifier; evaluate on sample pages.
3	Deploy element detectors and table recognizer; stand up Markdown/CSV writers.
4	Iterate on semantic assembly, run core evaluations, and prepare presentation materials.

Current progress: completed literature review, defined architecture, and collected representative notebook scans.

6. Risks and Mitigation

- **Limited Hand-Drawn Training Data**
  - Mitigation: remove the information extraction steps and keep only the classification.
- **Complex Page Layouts**
  - Mitigation: flag difficult cases for human review.
- **Lighting Noise**
  - Mitigation: use normalization targets, augment brightness/contrast, and monitor detector confidence.

7. Conclusion and Future Work

FieldChartOCR delivers a modular approach to handwritten chart understanding that bridges the gap between raw scans and analytic-ready datasets. The simplified plan concentrates on a single, reliable pipeline that extracts layout structure and exports Markdown and CSV artifacts. Future work will explore richer diagram types and collaborative review tooling once the MVP demonstrates consistent performance.

References

1. Liu, X., Chen, L., Wei, Y., Luo, Y., Zhang, H., & Zhou, S. (2021). ChartOCR: Data extraction from chart images via a deep hybrid framework. WACV 2021. https://openaccess.thecvf.com/content/WACV2021/papers/Luo_ChartOCR_Data_Extraction_From_Charts_Images_via_a_Deep_Hybrid_WACV_2021_paper.pdf

2. Savva, M., Kong, N., Chhoun, C., Agrawala, M., & Heer, J. (2011). ReVision: Automated classification, analysis, and redesign of chart images. UIST '11, 393–402. https://doi.org/10.1145/2047196.2047247

3. Jung, D., Kim, W., Song, H., Hwang, J.-I., Lee, B., Kim, B., & Seo, J. (2017). ChartSense: Interactive data extraction from chart images. CHI '17, 6706–6717. https://www.microsoft.com/en-us/research/wp-content/uploads/2017/02/ChartSense-CHI2017.pdf

4. Rowtula, V., Oota, S. R., & Jawahar, C. V. (2019). Towards automated evaluation of handwritten assessments. (Conference details unavailable). https://cvit.iiit.ac.in/images/ConferencePapers/2019/PID6008523.pdf

5. Huang, K.-H., Chan, H. P., Fung, Y. R., Qiu, H., Zhou, M., Joty, S., Chang, S.-F., & Ji, H. (2024). From pixels to insights: A survey on automatic chart understanding in the era of large foundation models. IEEE Transactions on Knowledge and Data Engineering. https://arxiv.org/pdf/2403.12027v4

6. Li, Z., et al. (2024). Improving handwritten mathematical expression recognition via integrating convolutional neural network with transformer and diffusion-based data augmentation. IEEE Access. https://www.researchgate.net/publication/380548640_Improving_Handwritten_Mathematical_Expression_Recognition_via_Integrating_Convolutional_Neural_Network_with_Transformer_and_Diffusion-Based_Data_Augmentation