Plant Engineering Department
P&ID Digitalisation Project

Internship Completion Report

Deep Das and Muzammil

October 23, 2025

Abstract

This report presents the successful development and deployment of a novel, commercially viable pipeline for the digitalisation of legacy Piping and Instrumentation Diagrams (P&IDs) at LTTS. The project delivers a unique solution that automates the conversion of scanned or PDF-based diagrams into structured, editable digital artefacts, leveraging state-of-the-art language models and robust engineering practices. The pipeline is designed for real-world impact, reducing manual effort, improving documentation quality, and enabling advanced analytics for plant engineering workflows.

The proposed model can be downloaded from Kaggle Model Card with all the codes made during the internship from Github.

1 Introduction

P&IDs are essential for describing the flow of materials, instrumentation logic, and safety systems in industrial plants. Traditionally, these diagrams are maintained as scanned images or PDFs, which are difficult to update, audit, or integrate with modern digital systems. Recognising this challenge, the internship focused on creating an end-to-end automated pipeline that transforms these legacy diagrams into high-quality LaTeX/TikZ code and natural-language documentation, tailored for LTTS's commercial needs.

2 Objectives

The primary objective was to build a pipeline that is not only technically robust but also commercially deployable. The solution was required to:

- Automate the conversion of P&ID PDFs into LaTeX/TikZ code using a reliable converter.
- Feed the generated LaTeX code into a large language model (LLM) fine-tuned specifically for LaTeX-to-natural language conversion.

- Ensure the LLM produces detailed, step-by-step instructions describing every node, connection, and graphical property (e.g., shape, direction, thickness, boldness) in the diagram.
- Verify connections by cross-referencing described line flows and node inputs/outputs, ensuring semantic fidelity and correctness.
- Employ advanced prompt engineering to maximise model performance and clarity, with iterative refinement based on real P&ID examples.
- Integrate two distinct models: a fine-tuned GPT-2 (uploaded to Hugging Face) and a Llama 3.1 8B model (under development), both adapted for this unique task.
- Use DeTikZify to convert natural-language instructions back into TikZ code, completing the round-trip digitalisation process.
- Compile the final TikZ code to produce digitalised P&ID images, ready for integration and further analysis.

3 Commercial Context and Impact

The pipeline directly addresses critical pain points in LTTS's Plant Engineering Division. By automating the digitalisation process, the solution:

The impact is substantial:

- **Labour cost reduction:** Weeks of manual transcription and diagram recreation are eliminated, delivering immediate return on investment for large document sets.
- **Quality and consistency:** Automated outputs ensure uniform notation, reduce human error, and enable auditable revision histories.
- **Scalability:** The pipeline runs efficiently on both cloud GPUs (Kaggle) and local workstations, with robust fallback to CPU, supporting rapid scaling to thousands of diagrams.
- **Vendor independence:** Built on open standards (TikZ, Hugging Face), the solution avoids vendor lock-in and remains adaptable as technology evolves.
- **Foundation for advanced analytics:** Digitalised diagrams become searchable, analysable, and integrable with downstream systems, opening new opportunities for compliance, anomaly detection, and process optimisation.

4 Methodology and Technical Innovation

The methodology developed in this project represents a significant advancement in the digitalisation of P&ID diagrams, introducing a multi-stage pipeline that is both technically novel and practically impactful. The process is designed to address the real challenges faced by plant engineering teams, such as the difficulty of updating scanned diagrams, the risk of manual errors, and the need for high-fidelity digital artefacts.

Pipeline Overview

The pipeline consists of the following key stages:

1. **PDF to LaTeX/TikZ Conversion:** Legacy P&ID diagrams, typically available as PDFs, are first converted into LaTeX/TikZ code using a dedicated converter. This step is crucial because it transforms static images into a structured, editable format that can be processed programmatically. For example, a complex valve arrangement in a PDF is translated into TikZ commands specifying node shapes, positions, and connections.

2. **LaTeX/TikZ to Natural Language via LLM:** The generated TikZ code is then fed into a large language model (LLM) that has been fine-tuned for this specific task. The LLM is prompted to produce detailed, step-by-step instructions describing every graphical element. For instance, the model might output: "Node A is a rectangle at (1,2) with a thick border; it is connected to Node B by a dashed line running horizontally." This level of detail is essential for documentation, verification, and downstream automation.

3. **Connection Verification and Semantic Integrity:** One of the most impactful innovations is the automated verification of connections. The pipeline cross-references the described line flows and node inputs/outputs, ensuring that every connection in the diagram is accurately captured. For example, if the LLM describes a line from Node A to Node B, the system checks that Node B's input matches the expected connection. This prevents errors that could arise from manual transcription and ensures the semantic integrity of the digitalised diagram.

4. **Prompt Engineering and Model Fine-Tuning:** Before fine-tuning the LLM, extensive prompt engineering was conducted to optimise the clarity and coverage of the generated instructions. Prompts were iteratively refined using real P&ID examples, ensuring that the model could handle complex scenarios such as multi-branch pipelines, instrumentation loops, and safety interlocks. This step was critical for achieving high-quality outputs and is a major reason for the project's success.

5. **Dual-Model Approach:** The pipeline employs two models: a fine-tuned GPT-2 (available on Hugging Face) and a Llama 3.1 8B model (under development). The GPT-2 model is lightweight and fast, making it suitable for low-resource environments, while the Llama model offers greater capacity for handling large, complex diagrams. This dual-model strategy is unique—prior to this project, no solution existed that could perform this task with either model.

6. **Natural Language to TikZ Synthesis (DeTikZify):** The instructions generated by the LLM are then processed by DeTikZify, a specialised model that converts natural-language descriptions back into TikZ code. This enables round-trip digitalisation, allowing diagrams to be regenerated, verified, and harmonised for style and accuracy. For example, a description like "A pump connected to a valve by a thick line" is translated into precise TikZ commands, which can be compiled to produce a digital image.

7. **Compilation and Artefact Generation:** Finally, the TikZ code is compiled using pdflatex to produce high-quality digital images of the P&ID diagrams. All

artefacts—including the source TikZ, generated instructions, regenerated code, and compiled images—are logged for traceability and audit compliance.

Practical Examples and Advantages

To illustrate the practical benefits of this methodology, consider the following scenarios:

Example 1: Error Reduction in Manual Transcription Traditionally, engineers would manually transcribe scanned diagrams into digital formats, a process prone to errors such as missing connections or incorrect symbol usage. With the proposed pipeline, the conversion and verification are automated. For instance, if a line is supposed to connect a pump to a valve, the system ensures that both endpoints are correctly described and matched, eliminating the risk of oversight.

Example 2: Enhanced Documentation and Auditability The natural-language instructions generated by the LLM provide exhaustive documentation of every element in the diagram. This is invaluable for audits, training, and process optimisation. For example, an auditor can review the generated instructions to verify that all safety interlocks are present and correctly implemented, without needing to interpret raw TikZ code.

Example 3: Scalability and Adaptability The dual-model approach allows the pipeline to scale across different environments. On a local workstation with limited resources, the GPT-2 model can process diagrams quickly. For larger, more complex diagrams, the Llama 3.1 8B model provides the necessary capacity. This flexibility ensures that the solution is commercially viable and adaptable to a wide range of use cases.

Example 4: Round-Trip Verification and Style Harmonisation By converting diagrams from TikZ to natural language and back again, the pipeline enables round-trip verification. This means that any changes or corrections made in the natural-language stage can be reflected in the regenerated TikZ code, ensuring consistency and harmonisation of style across all diagrams. For instance, if a particular symbol needs to be updated to meet new standards, the change can be made in the instructions and automatically propagated to the code.

In summary, the methodology introduced in this project not only automates a previously manual and error-prone process but also enhances the quality, consistency, and utility of digitalised P&ID diagrams. The technical innovations—especially the connection verification, prompt engineering, and dual-model strategy—set a new benchmark for the field and deliver substantial commercial value to LTTS.

5 Implementation Details

The pipeline is implemented in Python 3.10, leveraging the Hugging Face transformers library, bitsandbytes for quantisation, and system-level `pdflatex` for rendering. The main entry points are the `pipeline.py` script (for CLI-free execution on Kaggle) and the `local_pipeline.ipynb` notebook (for local GPU development and debugging). The pipeline is fully parameterised, supporting batch processing, model selection, and robust error handling for network faults and memory constraints. Outputs include the source TikZ snapshot, generated instructions, regenerated TikZ code, compiled PDF images, and a summary JSON for traceability.

6 Experimental Results

Extensive dry runs were conducted on sample P&ID TikZ assets, demonstrating that the pipeline reliably captures equipment tags, signal flows, and layout cues with high precision. The regenerated TikZ code is structurally sound, and the round-trip process enables effective verification and style harmonisation. Quantised model loading ensures that the end-to-end runtime remains efficient, with typical diagrams processed in under 10 minutes on Kaggle's T4 GPUs, including dependency installation and optional TeX compilation.

Due to the absence of a publicly available dataset specifically tailored for Piping and Instrumentation Diagrams (P&ID), we leveraged the `datikz-v2` dataset. This dataset comprises over 360,000 human-created TikZ graphics, making it the largest TikZ dataset to date. The dataset includes a diverse range of scientific illustrations, encompassing flowcharts, circuit diagrams, and other technical schematics, which share structural similarities with P&ID diagrams.

For training purposes, we utilized the full training split of `datikz-v2`, which contains 94,532 samples, and for evaluation, we used the corresponding test split of 442 samples. This dataset's comprehensive coverage and alignment with our diagrammatic needs made it an ideal candidate for training our model to understand and generate TikZ code from textual descriptions. Each entry in the dataset provides a TikZ source snippet and its intended semantic interpretation, enabling supervised fine-tuning of generative models. In our experiments, we leveraged the training split of `datikz-v2` to teach the model how to translate raw TikZ code into structured, human-readable instructions. This dataset forms a rich resource for evaluating round-trip accuracy, textual instruction generation, and the structural integrity of regenerated TikZ diagrams.

7 Challenges and Mitigations

The pipeline assumes accurate TikZ input, so provenance documentation and snippet validation were implemented to mitigate risks from imperfect PDF-to-TikZ conversion tools. Large TikZ files were chunked with overlap to maintain continuity without exceeding model context limits. Semantic fidelity was improved through tailored prompts and numbered outputs, ensuring clarity for complex instrumentation loops and safety features. Resource constraints were managed via 4-bit quantisation and optional full-precision loading, enabling deployment across GPUs with varying memory budgets. Network resilience was achieved by catching and retrying CAS service errors and DNS failures during model downloads, and the pipeline automatically falls back to CPU if GPU memory is exhausted, ensuring uninterrupted operation.

8 Future Work

Building on the success of this project, future enhancements could include integrating vision-based verification (e.g., DeTikZify scoring) to automatically compare compiled PDFs against original diagrams, developing active-learning loops where human corrections feed into continuous model fine-tuning, expanding support for instrumentation standards (ISA, ISO) by incorporating symbol ontologies and validation schemas, and exploring batching strategies for processing entire document sets with shared caches for model

weights and TikZ preprocessing artefacts.

9 Conclusion

This internship project has delivered a high-impact, commercially viable pipeline for the digitalisation of P&ID diagrams, setting a new standard for automation and documentation in plant engineering. By combining advanced language models, rigorous prompt engineering, and robust software design, the solution enables LTTS to modernise its documentation practices, reduce manual effort, and unlock new opportunities for analytics and process optimisation. The originality and effectiveness of the approach are evidenced by the successful deployment and the absence of any prior solution for this task.

10 References

- Potamides, AutomaTikZ (2024). `https://github.com/potamides/AutomaTikZ`
- Meta AI, Llama 3 model card (2024). `https://huggingface.co/meta-llama`
- DeTikZify: Semantics-preserving figure synthesis (2024). `https://arxiv.org/abs/2405.15306`
- DaTikZ-v2 dataset (2023). `https://huggingface.co/datasets/nllg/datikz-v2`