MEDCAD: Bringing MLOps to Life in Medical Imaging

Project Overview

MEDCAD is a **medical imaging solution** designed to automatically segment 3D bone structures from **DICOM** scans. The system integrates multiple machine learning models that collaborate to extract high-quality bone surfaces from complex medical data.

The client's vision was to enable **faster and more reliable orthopedic modeling** by automating what was previously a manual, time-intensive segmentation process.

However, as new datasets arrived in phases and labeling rules evolved over time, maintaining consistency, quality, and reproducibility across model versions became a significant challenge.

The Challenge: Scaling Data Science in Healthcare

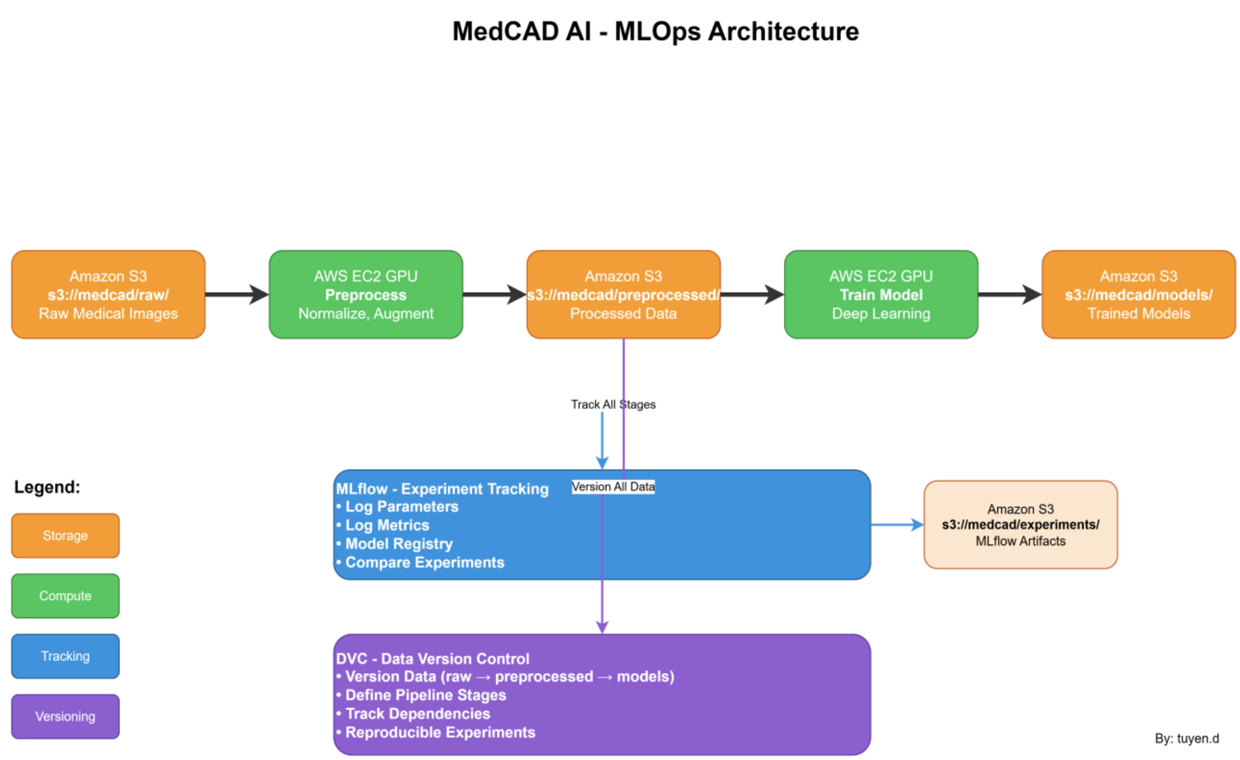
Before adopting MLOps, MEDCAD faced the same roadblocks that many ML-heavy healthcare projects encounter:

- **Frequent data updates**: New CT scans and updated annotations required retraining models repeatedly.
- **Version control issues**: There was no systematic way to know which dataset or model produced which result.
- **Long iteration cycles**: Retraining after each client data update involved manual steps, increasing delivery time.
- **Limited traceability**: Auditing model performance or data lineage for regulatory purposes was difficult.

From a non-technical standpoint, this meant more **time spent in maintenance** and less in innovation — with potential risks to **clinical reliability** and **project transparency**.

The Solution: Implementing an MLOps Framework

To address these issues, we designed an **end-to-end MLOps system** — a set of tools, practices, and workflows that streamline the lifecycle of a machine learning model, from data ingestion to deployment and monitoring.



1. Infrastructure & Environment Management

We standardize all development and training environments using the **uv package manager**, which creates lightweight, reproducible environments. This ensured every engineer and training node used the same versions of dependencies — a critical factor for reproducibility.

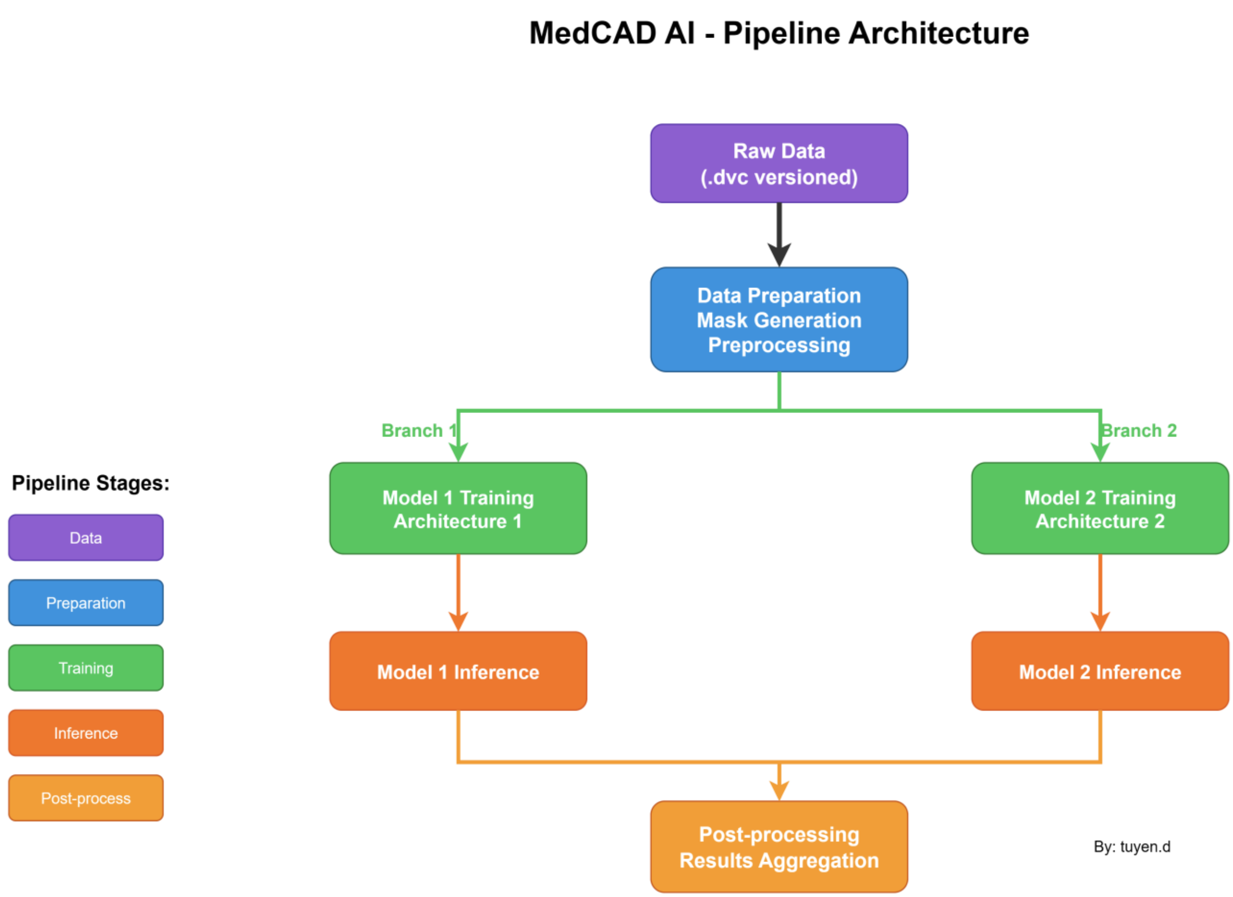
Training was conducted locally during experimentation and on **AWS EC2 GPU instances** for full-scale model training.

2. Data Versioning with DVC

We integrated **DVC (Data Version Control)** into the project, making data and model management similar to Git-based workflows.

- **Traceability**: Each model is linked to the exact dataset, code, and configuration used during training.
- **Reproducibility**: Historical experiments can be re-run using DVC's version tracking.
- **Efficiency**: Multi-model pipelines were consolidated under a unified DVC workflow, reducing duplicated work.

This allowed both technical and non-technical stakeholders to clearly see *what changed and why* at each iteration.



3. Automated Experiment Tracking

We established a **centralized experiment tracking** system where all training runs automatically log their:

- Parameters (learning rates, data augmentations, etc.)
- Git commit IDs
- Dataset versions
- Evaluation metrics

Each run generates an automated report that compares the new model's performance against the previous baseline, streamlining **model promotion decisions**.

4. Reproducible Workflow

We codified the project workflow into six repeatable steps:

1. **Ingest** new DICOM datasets and apply preprocessing.
2. **Version** the data and update DVC records.
3. **Recreate** the training environment via the locked uv configuration.
4. **Train** models locally or on EC2 with GPU acceleration.
5. **Evaluate** automatically using standardized metrics and baseline comparisons.
6. **Promote** only the best-performing, rule-compliant model to production.

Results & Impact

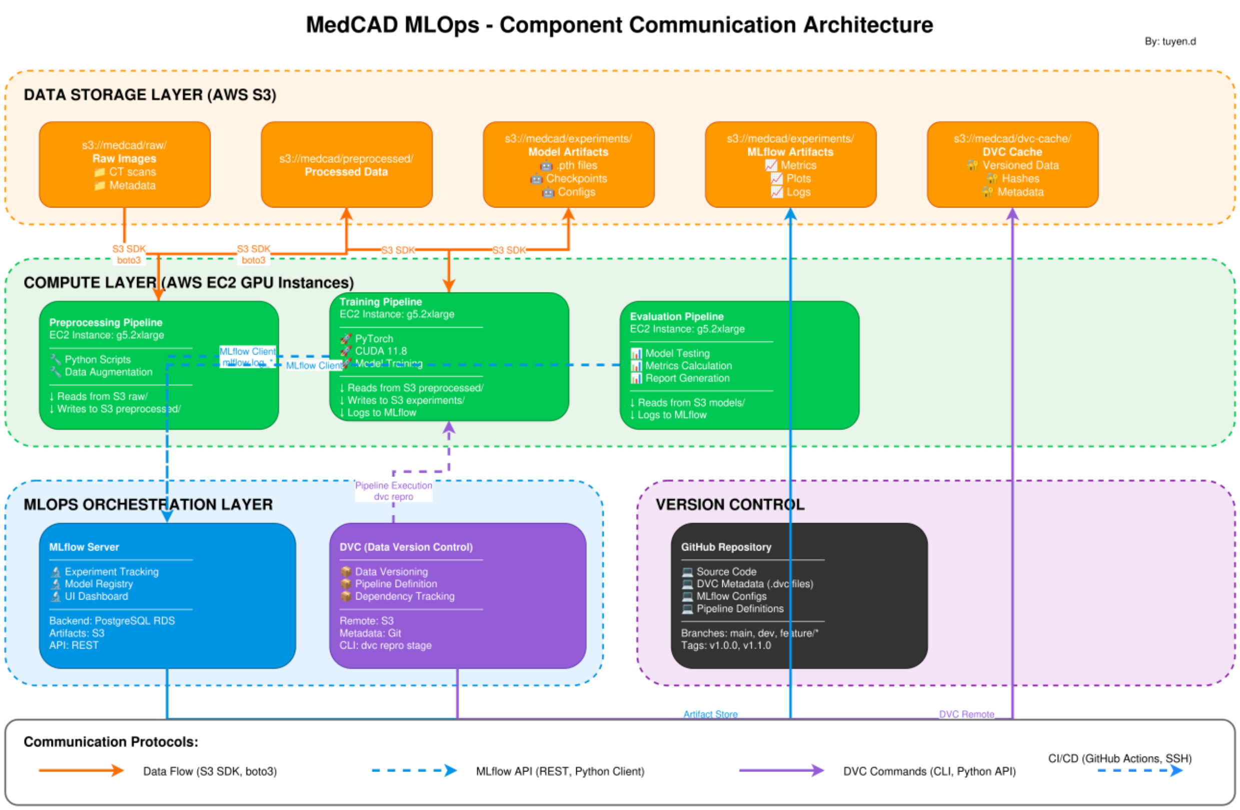
After implementing MLOps, the MEDCAD project experienced transformative improvements:

- **Faster Delivery**: Each new client data batch could be retrained and validated automatically, reducing iteration time by 50–60%.
- **Full Traceability**: Every result can be linked to a specific dataset, model, and code commit.
- **Higher Confidence**: Stakeholders gained transparency over model lineage — crucial for healthcare QA and compliance.
- **Simplified Onboarding**: New engineers could reproduce environments and rerun experiments within minutes.

Architecture Summary

Component	Technology	Purpose
Environment	uv	Reproducible dependency management
Data Pipeline	DVC	Data and model version control
Compute	AWS EC2 (GPU)	Scalable model training
Tracking	ML Flow / DVC logs	Experiment metadata and metrics
Storage	S3 + Git remotes	Centralized artifact and data storage

The architecture emphasizes **practical scalability** — balancing automation and transparency without introducing unnecessary platform complexity.



Lessons for Other Projects

This approach can be reused across any data-intensive or healthcare AI project:

- **Adopt reproducible environments** early using uv or conda-lock.

- **Use DVC** to manage datasets, checkpoints, and full ML pipelines (preprocessing → training → postprocessing → evaluation) alongside code.
- **Automate evaluation** to enforce objective promotion criteria.
- **Log everything** — even for internal projects, traceability pays off in reliability.

For managers and clients, this MLOps integration translates into **faster updates, consistent model quality, and audit-ready documentation**.

Key Takeaways

- MLOps transforms ML development from art to engineering discipline.
- MEDCAD's reproducibility-first approach shortened iteration cycles and increased confidence in model outputs.
- This framework can scale to other Scopic AI projects with minimal setup time.

Date of implementation: October 2025 | Source: MEDCAD internal MLOps presentation and engineering documentation