# Can Coding Agents Build Robust Baselines? A Skill-Based Approach for Automating the Medical Imaging Model-Development Pipeline

Eugenia Moris[1] and José Ignacio Orlando[1,2]

[1] Arionkoder LLC, Boston, MA 02114, United States
eugenia.moris@arionkoder.com
[2] Yatiris, PLADEMA, UNICEN-CONICET, Tandil, Buenos Aires, Argentina

**Abstract.** Developing competitive deep learning baselines for medical imaging remains a highly iterative process requiring literature review, implementation, experimentation, and expert refinement. Existing automation approaches typically optimize isolated components, such as architecture search or hyperparameter tuning, rather than the complete baseline development process. We present an agentic AI Scientist workflow that combines literature-guided reasoning, automated code generation, and hypothesis-driven experimentation to generate competitive baseline models for medical imaging challenges. The framework is evaluated on four public benchmarks spanning segmentation, classification, and detection. Across all tasks, the Experimentation Pipeline consistently improves validation performance, achieving competitive leaderboard results, including 6th place on both PUMA tracks (15 teams) and 31st place on MILK10k (125 teams). On MIDOG25, the resulting model also demonstrates strong domain generalization across scanners, tumor types, and species. Using the same workflow across all challenges without task-specific redesign, we demonstrate that skill-based, literature-guided agentic workflows can substantially reduce the engineering effort required to develop competitive medical imaging baselines.

**Keywords:** AI Scientist · Medical Image Analysis · Agentic Workflows

## 1 Introduction

Deep learning has become the dominant paradigm for medical image analysis, achieving strong performance in segmentation, classification, and detection [12]. However, developing a competitive baseline for a new problem remains an iterative and expertise-intensive process. It requires numerous interdependent decisions spanning preprocessing, model architecture, optimization, data augmentation, loss design, and evaluation [9, 11, 12, 15]. These decisions require both clinical and machine learning expertise, making baseline development costly, time-consuming, and difficult to scale [6, 8, 10].

Previous efforts have addressed isolated stages of this workflow, including hyperparameter optimization [8], neural architecture search [16, 20], and automated data augmentation [4,5]. More recently, LLM-based agents have extended

automation to literature analysis, code generation, experiment execution, and scientific workflows [3,13,14,17,18]. Nevertheless, existing approaches either optimize isolated stages of model development or provide general-purpose research assistants rather than workflows tailored to medical imaging [4,8,13]. We instead investigate whether literature-guided planning, automated implementation, and evidence-driven experimentation can be integrated into a unified workflow for automatically developing competitive medical imaging baselines with human supervision.

Experienced researchers review the literature before allocating computational resources, using prior evidence to understand the clinical problem, identify promising approaches, and avoid uninformative experiments. Motivated by this process, we propose an agentic AI Scientist workflow that combines literature-guided planning, automated implementation, and hypothesis-driven experimentation through reusable skills. A persistent experiment tree records both positive and negative experimental evidence, improving transparency and reproducibility. We evaluate the framework on four public medical imaging challenges spanning segmentation, classification, and object detection. Without task-specific redesign, the workflow adapts to heterogeneous tasks, achieves competitive leaderboard performance, and produces strong baselines within 2–7 days. Together, these results demonstrate that literature-guided, evidence-driven AI Scientist workflows can rapidly produce competitive and reproducible medical imaging baselines across diverse tasks.

## 2 Methods

Figure 1 presents the proposed AI Scientist workflow. Starting from a labeled dataset, a Data Description skill generates a structured task specification, while a Data Preparation skill converts the original files into a standardized format used by the subsequent pipelines.

The workflow consists of three sequential pipelines. The *State-of-the-Art (SOTA) Knowledge Pipeline* reviews the literature to generate an initial development plan and identify reusable framework components. The *Model Setup Pipeline* automatically implements the required data, model, optimization, and evaluation modules. Finally, the *Experimentation Pipeline* iteratively proposes, executes, and analyzes experiments, updating a persistent experiment plan and experiment tree that guide subsequent decisions within an available computational budget.

Each pipeline is composed of independent LLM-driven skills with explicit inputs and outputs. The resulting artifacts provide a transparent and reproducible record of the complete baseline-development process while keeping the researcher in the loop through lightweight review checkpoints throughout the workflow.

### 2.1 SOTA Knowledge Pipeline

The SOTA Knowledge Pipeline (Figure 1, left) transforms a raw dataset into an initial development plan by combining problem characterization, literature

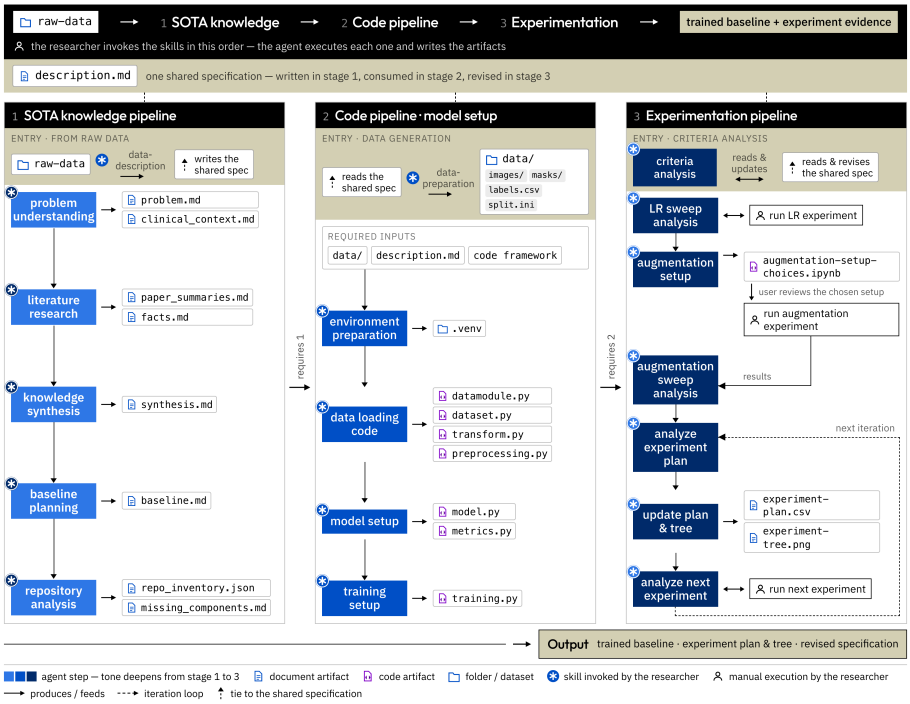



Fig. 1: Overview of the proposed skill-based agentic framework. Raw inputs are converted into a structured task description and a standardized dataset. Three phases then synthesize task-specific knowledge, construct an executable modeling pipeline, and iteratively select, execute, and analyze experiments while maintaining a persistent experiment plan and tree. The task description persists across all phases and the researcher invokes each skill in order, reviewing and running the experiments the framework proposes.

analysis, and repository inspection. First, the framework analyzes the task from both clinical and machine learning perspectives, identifying the key challenges, dataset characteristics, evaluation protocol, and potential sources of variability. These observations define the research questions used to guide the subsequent literature review.

Relevant publications and challenge reports are then synthesized to compare evidence across the main design axes, including model architecture, optimization, data augmentation, loss functions, and evaluation. Based on this evidence, the pipeline generates an initial development plan that specifies the baseline configuration. Uncertain or conflicting recommendations are explicitly formulated as experimental hypotheses for the Experimentation Pipeline, while repository analysis identifies the components that can be reused and those requiring implementation.

### 2.2 Model Setup Pipeline

The Model Setup Pipeline (Figure 1, middle) transforms the literature-guided development plan into an executable training framework. Based on the generated plan and repository analysis, it configures the execution environment, implements the dataset-specific loading and preprocessing modules, and instantiates the model architecture, optimization strategy, loss function, and evaluation metrics. The resulting components are integrated into the training framework, producing a reproducible baseline implementation that serves as the starting point for the Experimentation Pipeline. Data augmentation is intentionally excluded, as its policy and parameters are optimized during experimentation.

### 2.3 Experimentation Pipeline

The Experimentation Pipeline (Figure 1, right) transforms the initial development plan into a competitive baseline through iterative, evidence-driven experimentation. Rather than optimizing a predefined hyperparameter space, it follows a scientific workflow that defines evaluation criteria, tests hypotheses, interprets results, and prioritizes subsequent experiments by expected information gain within a limited computational budget. Evaluation begins by establishing the challenge metric together with secondary metrics for clinically relevant or underrepresented classes. Initial experimental evidence is collected through learning-rate calibration and RandAugment-based policy optimization [5]. Rather than searching unrestricted augmentation parameters, the pipeline first estimates clinically plausible parameter ranges for each candidate transformation using image-based inspection and literature-derived constraints, ensuring that only realistic image variations are explored during policy optimization. Subsequent experiments are generated through a structured process: rule-based logic selects the next component to evaluate, reusable templates define the experiment structure, and the LLM instantiates the hypothesis, configuration, and validation criterion, prioritizing literature-supported alternatives when available. Each experiment modifies a single design decision whenever possible, and completed experiments update a persistent experiment plan and tree that guide future iterations by recording accepted and rejected evidence.

## 3 Experimental Setup

### 3.1 Implementation details

The framework is orchestrated through Claude Code [1], with all reasoning and code generation performed by Claude Sonnet 4. Each skill is implemented as a task-specific prompt template defining its objective, inputs, outputs, and operational constraints. Rather than relying on a single monolithic prompt, the workflow decomposes the development process into specialized skills that exchange structured Markdown, YAML, and CSV artifacts. The implementation is based

Table 1: Summary of the benchmark challenges used to evaluate the workflow.

| Challenge | Task | Train/Val/Test | Metric | Evaluation |
|---|---|---|---|---|
| PUMA | Segmentation + Detection | 174 / 31 / Hidden | Dice / Macro-F1 | Leaderboard |
| MILK10k | Classification | 4454 / 786 / Hidden | Macro-F1 | Leaderboard |
| MIDOG25 | Detection | 8626 / 1747 / 1564 | Object F1 | Held-out test |

on a shared PyTorch Lightning [7] framework providing the common training infrastructure. The Model Setup Pipeline automatically generates the task-specific components—including the data module, model, loss functions, augmentations, optimizer, and evaluation configuration—within this reusable framework. Experiments were executed on two hardware platforms. PUMA tissue segmentation was developed on Apple Silicon MPS backend. PUMA nuclei detection and MILK10k were trained on an NVIDIA RTX 4090 (24 GB), while MIDOG25 initially used MPS before transitioning to the RTX 4090 for foundation-model experiments requiring additional GPU memory.

Although the workflow automates literature synthesis, code generation, experiment planning, and configuration, human supervision remains part of the development process. After each stage, the researcher reviews the generated artifacts, validates the proposed implementation, launches the suggested experiments, and monitors their execution. Manual modifications to the generated modeling code were not required in the reported experiments. Instead, the reusable framework enables the agent to generate only the task-specific components while reusing a common training, optimization, logging, and evaluation infrastructure. The researcher's role shifts from implementing and configuring models to supervising the development process and validating the generated evidence, preserving human oversight while substantially reducing engineering effort.

### 3.2 Materials and evaluation

The proposed workflow was evaluated on four benchmark tasks derived from three public medical imaging challenges covering semantic segmentation, multi-class classification, and object detection (Table 1). The workflow first searches for patient-, specimen-, or slide-level identifiers and generates group-aware data splits whenever available. Otherwise, it uses reproducible image-level random splitting. As PUMA and MILK10k provide no such metadata, we created 85/15 image-level splits (seed 42). MIDOG25 instead provides whole-slide identifiers, enabling grouped train, validation, and held-out test splits.

For all challenges, model selection relied exclusively on validation performance using the official challenge metrics. For MIDOG25, we report both Macro-$F_1$ and $F_{1,\mathrm{AMF}}$, the F1-score of the atypical mitotic figure class. The Experimentation Pipeline recorded the validation metric after learning-rate calibration, augmentation optimization, and experiment exploration. Across all three stages, the total number of training runs was capped at 100 per challenge. Final models

Table 2: Performance across challenges with open leaderboard evaluations. Validation scores are reported after learning-rate (LR) selection, data augmentation, and experiment exploration. Test scores include results of our final baseline vs. the most similar method submitted and the leading model.

| Challenge | Metric | Validation | | | Test | | | | Time to model (days)† |
|---|---|---|---|---|---|---|---|---|---|
| | | LR selection | Data augmentation | Experiment exploration | Final baseline (ours) | Similar method | First-place model | Rank | |
| PUMA Tissue segmentation | Macro Dice | 0.672 | 0.675 | 0.732 | **0.691** | 0.554 (+0.137) | 0.783 (−0.092) | 6/15 | 2 |
| PUMA Nuclei detection | Macro-F1 | 0.459 | 0.467 | 0.717 | **0.624** | 0.586 (+0.038) | 0.658 (−0.034) | 6/15 | 4 |
| MILK10k ISIC Challenge | Macro-F1 | 0.474 | 0.493 | 0.528 | **0.490** | 0.515* (−0.025) | 0.609* (−0.119) | 31/125* | 5 |
| MIDOG25 Atypical mitosis classification | F1 | 0.757 | 0.791 | 0.855 | — ‡ | 0.882 | 0.907 | — | 7 |

* Among methods that do not use external datasets.
† Elapsed project duration measured from the first executed training run to the final selected baseline; not researcher working hours.
‡ A direct comparison is not possible because our model was evaluated on a different test set.

for PUMA and MILK10k were evaluated through the official challenge servers, whereas MIDOG25 was evaluated on the independent held-out test split. We additionally compare our results with the most similar published approach and, when available, the top-ranked challenge submission.

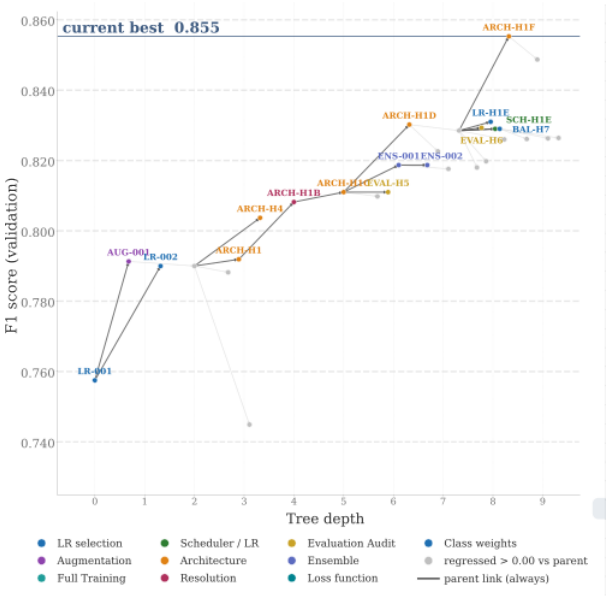
## 4 Results and Discussion

Before experimentation, the SOTA Knowledge Pipeline generated an initial development plan by synthesizing evidence from the literature and challenge documentation. The resulting plan specified the initial model architecture, optimization strategy, input resolution, loss function, augmentation policy, evaluation protocol, and prioritized experimental hypotheses.

Across all challenges, the Experimentation Pipeline validated many literature-derived decisions—including optimization strategies, augmentation policies, and several training configurations—while revising others, most notably the model architecture. These findings suggest that literature-guided planning provides a strong starting point, while systematic experimentation remains essential for adapting the solution to the characteristics of a specific dataset.

Table 2 quantifies the effect of the refinement process. Validation performance improved after each stage of the Experimentation Pipeline across all four tasks, with the largest gains obtained through experiment exploration. The final models also performed competitively on independent leaderboard evaluations, ranking 6th of 15 teams in both PUMA tracks and 31st of 125 submissions on MILK10k. Although leaderboard comparisons are not controlled, these results suggest that the proposed workflow generalizes across diverse medical imaging tasks.

Figure 2 illustrates how the Experimentation Pipeline incrementally refines the initial development plan through evidence-driven experimentation. Across 30 MIDOG25 experiments, the best validation $F_1$ improved from 0.757 to 0.855. The framework evaluated hypotheses spanning optimization, data augmentation, architecture, input resolution, and training strategies while recording both




| Experiment | F1-score (val) | Accepted |
|---|---|---|
| LR-001 | 0.757 | ✓ |
| AUG-001 | 0.791 | ✓ |
| LR-002 | 0.790 | ✓ |
| FULL-001 | 0.790 | ✗ |
| SCH-001 | 0.788 | ✗ |
| ARCH-H1 | 0.792 | ✓ |
| AUG-H2 | 0.745 | ✗ |
| ARCH-H4 | 0.804 | ✓ |
| ARCH-H1B | 0.808 | ✓ |
| ARCH-H1C | 0.811 | ✓ |
| SCH-H1C | 0.810 | ✗ |
| EVAL-H5 | 0.811 | ✓ |
| ENS-001 | 0.819 | ✓ |
| ARCH-H1D | 0.830 | ✓ |
| ENS-002 | 0.819 | ✓ |
| ENS-004 | 0.823 | ✗ |
| ENS-003 | 0.818 | ✗ |
| ARCH-H1E | 0.829 | ✗ |
| LOSS-H3 | 0.818 | ✗ |
| EVAL-H6 | 0.829 | ✓ |
| ENS-005 | 0.820 | ✗ |
| LR-H1E | 0.831 | ✓ |
| SCH-H1E | 0.829 | ✓ |
| BAL-H7 | 0.829 | ✓ |
| SCH-H1F | 0.826 | ✗ |
| ARCH-H1F | 0.855 | ✓ |
| BAL-H7B | 0.826 | ✗ |
| BAL-H7E | 0.849 | ✗ |
| BAL-H7C | 0.826 | ✗ |
| BAL-H7D | 0.826 | ✗ |

Fig. 2: Experimentation workflow on MIDOG25. Left: experiment tree showing validation $F_1$ progression, with nodes colored by decision category and gray nodes indicating rejected experiments. Right: chronological sequence of executed experiments. The final accepted configuration achieved the highest validation score ($F_1 = 0.855$).

successful and unsuccessful experiments. Rather than being discarded, rejected experiments are preserved as negative evidence that guides subsequent decisions. Overall, 17 experiments were accepted and 13 were rejected, allowing the search to branch from the strongest validated configuration while avoiding previously explored directions. Unlike conventional experiment tracking systems such as MLflow and Weights & Biases, which primarily organize and record experimental artifacts and metadata, the proposed experiment tree preserves structured positive and negative evidence to directly inform future experimentation [2, 19].

Table 3 summarizes the domain generalization of the automatically generated MIDOG25 baseline. Validation and held-out test performance remained consistent, with similar results across species, scanners, and tumor types despite substantial biological and acquisition variability. Although evaluated on a held-out split derived from the released challenge data, these findings suggest that literature-guided planning and evidence-driven experimentation can produce robust baselines that generalize across heterogeneous medical imaging domains.

### 4.1 Limitations

This study evaluates the integrated workflow rather than the contribution of its individual components. We therefore do not isolate the effect of the SOTA

Table 3: Evaluation of MIDOG25. Left: overall validation and test performance. Right: summary of domain generalization across heterogeneous domains.

**Overall**

| Metric | Val. | Test |
|---|---|---|
| Macro-$F_1$ | 0.8553 | 0.8473 |
| $F1_{AMF}$ | 0.7579 | 0.7457 |

**Domain Generalization**

| Domain | Min | Max | Mean |
|---|---|---|---|
| Species | 0.731 | 0.764 | 0.748 |
| Tumor types | 0.633 | 0.880 | 0.771 |
| Scanners[†] | 0.694 | 0.807 | 0.756 |

[†] Excluding Hamamatsu S360 scanner (1 AMF).

Knowledge Pipeline, the Model Setup Pipeline, or the Experimentation Pipeline through ablation studies. In addition, experiments were performed using a single random seed, no confidence intervals are reported, and we did not compare against human-developed baselines, conventional hyperparameter optimization, or existing AutoML systems under identical computational budgets. Consequently, the reported results should be interpreted as evidence for the effectiveness of the complete workflow rather than of its individual modules. Future work should benchmark against established AutoML and augmentation-search methods (e.g., auto-sklearn, auto-augment).

Similar patterns emerged across the evaluated challenges. Architecture exploration consistently produced the largest performance gains, whereas loss-based class reweighting and learning-rate schedulers were never retained. Performance on underrepresented classes was instead improved through oversampling in PUMA and class-specific decision thresholds in MILK10k. Although these are descriptive observations rather than controlled ablations, they show that the experiment tree captures recurring optimization patterns across tasks.

Finally, the workflow operates within a predefined orchestration strategy and reusable training framework. Although the agent iteratively refines development decisions using experimental evidence, it cannot autonomously redesign the workflow, revisit the literature in response to unexpected findings, or introduce new development stages. Enabling more adaptive scientific reasoning while preserving transparency, reproducibility, and human oversight remains an important direction for future AI Scientist systems. Future studies will investigate more autonomous scientific reasoning inspired by recent AI Scientist systems [13, 14].

## 5 Conclusion

We presented an agentic framework that helps develop competitive baseline models for medical imaging challenges. The workflow combines literature-guided planning, automated implementation, and evidence-driven experimentation to transform raw data into a trained baseline with lightweight human oversight.

We evaluated the framework on four public medical imaging challenges spanning segmentation, classification, and detection. Across all tasks, the Experimentation Pipeline consistently improved validation performance, while the result-

ing automatically generated baselines achieved competitive leaderboard performance. The MIDOG25 results further demonstrated that the generated baseline generalized without requiring dedicated domain-adaptation strategies.

These findings demonstrate that a structured AI Scientist workflow can automate a substantial portion of the machine learning development process for medical imaging while maintaining competitive performance across diverse tasks. More broadly, we believe this study is a step toward AI systems that support researchers throughout the model development lifecycle — letting experts focus on scientific reasoning, hypothesis generation, and clinical interpretation rather than repetitive implementation — and that skill-based AI Scientists represent a practical path toward scalable, reproducible machine learning research in medical imaging.

**Acknowledgments.** The authors thank Arionkoder for supporting this research and providing the computational resources used in this study.

**Disclosure of Interests.** The authors declare that they have no competing interests relevant to the content of this article.

## References

1. Anthropic. Claude, 2026. Accessed: 2026-07-30.
2. Lukas Biewald. Experiment tracking with weights and biases. https://www.wandb.com/, 2020.
3. Kyle Chan et al. MLE-Bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations (ICLR)*, 2025.
4. Ekin D. Cubuk et al. Autoaugment: Learning augmentation strategies from data. In *CVPR*, pages 113–123, 2019.
5. Ekin D. Cubuk et al. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, pages 702–703, 2020.
6. Andre Esteva et al. Deep learning-enabled medical computer vision. *npj Digital Medicine*, 4(1):5, 2021.
7. WA Falcon. Pytorch lightning. *GitHub*, 3, 2019. https://github.com/Lightning-AI/pytorch-lightning.
8. Matthias Feurer et al. Efficient and robust automated machine learning. In *NeurIPS*, volume 28, 2015.
9. Fabian Isensee et al. nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021.
10. Christopher J. Kelly et al. Key challenges for delivering clinical impact with artificial intelligence. *BMC Medicine*, 17(1):195, 2019.
11. Dominik Kreuzberger et al. Machine learning operations (MLOps): Overview, definition, and architecture. *IEEE Access*, 11:31866–31879, 2023.
12. Geert Litjens et al. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017.
13. Chris Lu et al. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

14. Samuel Schmidgall et al. Agent laboratory: Using LLM agents as research assistants. *arXiv:2501.04227*, 2025.
15. D. Sculley et al. Hidden technical debt in machine learning systems. In *NeurIPS*, volume 28, 2015.
16. Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.
17. Guanzhi Wang et al. Voyager: An open-ended embodied agent with large language models. *arXiv:2305.16291*, 2023.
18. Lei Wang et al. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*, 2023.
19. Matei Zaharia et al. Accelerating the machine learning lifecycle with MLflow. *IEEE Data Engineering Bulletin*, 2018.
20. Barret Zoph and Quoc Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017.