2025 IEEE International Conference on eScience (eScience) | USGov | DOI: 10.1109/eScience65000.2025.00061

EvoAgents: Generative AI for Rapid Architecture Search in Cell Nuclei Detection

Samantha Alonso
College of Computing, Data Science, and Society
University of California, Berkeley
Berkeley, USA
salonso@berkeley.edu

André Bauer
College of Computing
Illinois Institute of Technology
Chicago, USA
abauer7@iit.edu

***Abstract*—Science relies on image data across several disciplines like microbiology, but developing effective deep learning models for segmentation and detection tasks present challenges, including limited time, expertise, or financial resources. We introduce an evolutionary agent that autonomously generates, evaluates, and evolves convolutional neural networks (CNNs) based on natural language descriptions. Powered by large language models (LLMs) and guided by a genetic algorithm, our agent allows scientists to obtain trainable, reproducible model code for their specific tasks without manual development. A case study on cell nuclei detection demonstrates the pipeline's feasibility and therefore establishes a baseline for architectures driven by LLMs. Designed for generalizability to diverse scientific imaging problems, this agent acts as a collaborative interface between deep learning and science and therefore contributing to scientific discovery and reduced economic and technical barriers.**

***Index Terms*—Generative AI Agents, Large Language Models, Neural Architecture Search, Image Segmentation, Cellular Microbiology**

I. Introduction

Deep neural networks have revolutionized science that relies on image data, enabling breakthroughs in microbiology, medical imaging, and remote sensing. Yet even today, designing a high-performing CNN typically requires expert intuition and extensive hyper-parameter tuning. Traditional reinforcement-learning NAS (neural architecture search) [1] and Automated Machine Learning (AutoML) pipelines remain restrictive for many labs. Evolutionary NAS methods alleviate some cost [2] but still require specialized tooling, and domain-specific frameworks for cellular image analysis often depend on manually encoded expert knowledge [3].

Recently, agentic multi-step systems that use LLM reasoning with automated search—e.g. AlphaEvolve [4] and SELA [5]—have shown that natural-language prompts can yield executable code that improves through iterative feedback. Inspired by these works, we propose an agentic NAS co-pilot that lets domain scientists describe a task (e.g. "segment nuclei in my 250-image dataset") and receive a refined, reproducible convolutional neural network (CNN) without manual architecture design.

Image analysis of cells and tissues is critical for numerous biomedical applications. We showcase our agentic framework by testing its ability to autonomously generate convolutional neural networks (CNNs) for cell nuclei detection in microscopic images. Our framework reduces best-case validation loss by 65% on a custom dataset, achieving a Dice coefficient of 0.51–0.53 and Intersection over Union (IoU) of 0.40–0.42.

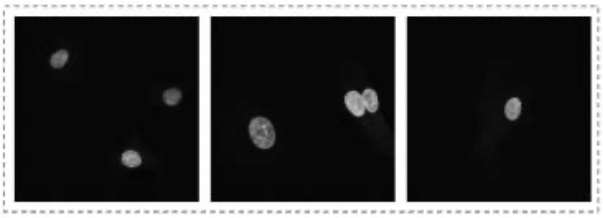

Fig. 1. Sample microscopic images of cell nuclei.

II. Methodology and Implementation

Our pipeline consists of two cooperating agents (Figure 2): An LLM-based *Architect* that emits candidate PyTorch models and a Genetic-Algorithm *Optimizer* that evolves those models over generations—a strategy similar to AlphaEvolve's code-diff evolution but applied to the structure and organization of a neural network.

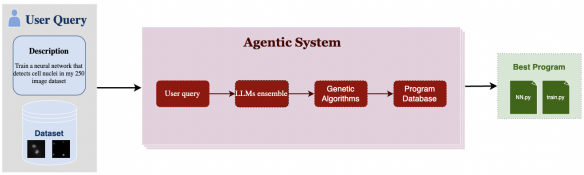

Fig. 2. Overview of the agentic system architecture. Users provide natural language descriptions of their imaging task and dataset. The system processes queries through an LLM ensemble, applies genetic algorithms for optimization, and maintains a program database to generate the best performing neural network implementation.

A. LLM-Driven Model Generation

The *Architect* samples a variety of CNN guidelines given a text query and simple dataset description. An initial population

DOI 10.1109/eScience65000.2025.00061

of three candidate architectures is generated by three different LLMs. Each candidate is tested on the target dataset for a small number of epochs (see details in Section II-B). The top two performing models from this initial pool are then selected to serve as the parent architectures for the evolutionary search.

B. Evolutionary Refinement

Each sampled model is trained for five epochs, recording the validation loss, Dice, and Intersection over Union (IoU) or other measures depending on the task. These measures serve as multi-objective fitness. The following three actions operate on serialized model code, while a simulated-annealing schedule tempers selection pressure to avoid premature convergence:
Selection: The two best performing candidate models from the population are chosen as parents for the next generation.
Crossover: A new model is created by combining the architectural components (e.g., number and type of layers, kernel sizes) of two parent architectures.
Mutation: A new model is created by randomly tweaking a parameter or structure of a parent network, such as changing a kernel size or inserting a new layer.

We maintain a population of three programs per generation, running for ten generations with fitness measured using validation cross-entropy loss.

III. Experiments and Results

To validate our framework, we constructed a collection of 250 grayscale fluorescence microscopy images of cultured cells with manually annotated nuclei masks. This was done by following the data preprocessing protocols outlined by Feng et al. [3]. The images were resized to 224×224 pixels and randomly split into 200 training and 50 validation samples.

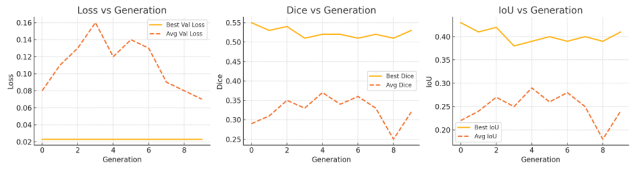

Fig. 3. Evolution of model performance across 10 generations of evolutionary search. (Left) Validation loss showing steady improvement, dropping from 0.067 to 0.023 (65% reduction). (Middle) Dice coefficient progression demonstrating consistent segmentation quality (0.51–0.53). (Right) Intersection over Union (IoU) metrics showing stable convergence around generation 6–7 (0.40–0.42).

Figure 3 shows convergence curves: best-case loss drops from 0.067 to 0.023 (65% reduction), while Dice and IoU rise steadily. The validation loss (left panel) demonstrates clear improvement, with the average population loss decreasing from 0.08 to 0.07. The Dice coefficient (middle panel) shows the best model maintaining consistent performance around 0.51-0.53, and the IoU metric (right panel) achieves approximately 0.40-0.42, competitive for cell nuclei segmentation tasks.

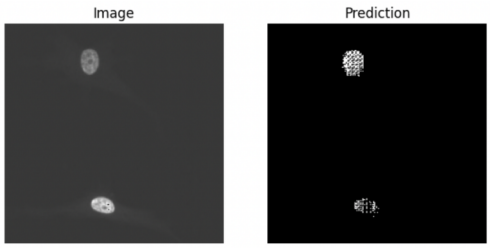

Fig. 4. Original fluorescence microscopy image of cell nuclei (Left) vs. predicted segmentation mask from the top-performing model (Right).

Figure 4 provides qualitative results from the best model architecture. Architecture evolution is interpretable: early generations add skip connections; later mutations fine-tune kernel sizes and channel widths. The evolutionary search converged after approximately 6-7 generations, with training time averaging 10 minutes per generation on standard GPU hardware.

IV. Conclusion and Future Work

We demonstrate that an agentic, LLM-driven NAS autonomously generates trainable convolutional neural networks (CNNs) for scientific applications with minimal human intervention. Our case study on cell nuclei detection validates the feasibility of our pipeline, achieving baseline segmentation (65% validation loss reduction, Dice 0.51–0.53, IoU 0.40–0.42) without manual architecture design. This approach aims to make deep learning accessible to domain scientists without technical expertise to develop effective neural networks.

Our next steps involve testing the agent on a variety of new scientific datasets that represent real world use cases, in order to ensure the system can perform reliably beyond the scope of this case study. We also plan to benchmark EvoAgents against established CNNs, such as U-Net, and compare its convergence speed and performance to architectures optimized by humans in order to enhance its competitiveness and further validate EvoAgents efficiency.

Acknowledgments

This work is supported in part by the National Science Foundation OAC-2150500 award. I would like to thank Dr. Yang Ha from Lawrence Berkeley National Laboratory (LBNL) for his input and for providing user validation throughout this research.

References

[1] B. Zoph and Q. V. Le, "Neural Architecture Search with Reinforcement Learning," in *Int. Conf. Learning Representations (ICLR)*, 2017.
[2] Y. Liu et al., "A Survey on Evolutionary Neural Architecture Search," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6550-6570, 2022.
[3] R. Feng et al., "Knowledge-Integrated Deep Learning Framework for Cellular Image Analysis in Parasite Microbiology," *STAR Protocols*, vol. 4, p. 102452, 2023.
[4] A. Novikov et al., "AlphaEvolve: A Coding Agent for Scientific and Algorithmic Discovery," Google DeepMind White Paper, 2025.
[5] Y. Chi et al., "SELA: Tree-Search Enhanced LLM Agents for Automated Machine Learning," Preprint, 2024.