**Cell Morphology Analyzer**

Paige Ring, Catrina Janos, Julia Heroux

**Introduction**

Cells are composed of several organelle compartments with distinct physical and chemical features that are essential to cellular function.2 Physical changes that occur in these organelle compartments leads to several forms of nonapoptotic regulated cell death (RCD). These changes can affect cellular signaling and have been implicated in being both beneficial and detrimental to diseases such as Alzheimer’s and cancer. RCD pathways affect cell morphology, and many changes that occur throughout the process. Morphology pertains to the study of the form and structure of cells and provides crucial insights into a variety of biological processes, including mechanisms of RCD. Apoptosis, necroptosis, and other types of cell death manifest through distinct morphological changes observable under a microscope. Fluorescent microscopy emphasizes the fundamental changes that occur within these RCD pathways. Traditionally, identifying these changes has required expert human analysis, a time-consuming and subjective process that can take weeks to analyze image data sets. However, with advances in artificial intelligence, particularly large language models (LLMs) capable of multimodal reasoning, it is now possible to automate and enhance this task.

Large language models (LLMs) can be used in many areas of research or day to day use. LLM’s are used for code generation, efficiency, content creation, automated tasks, and more. Connecting biology and the use of LLM’s is a newer area of interest as the world of AI is blooming.1 LLMs can help with image processing, code and imaging automation, and even performance evaluations for its own outputs. For example, some of the LLMs that were investigated included ChatGPT-4, Gemini-pro, and Claude. These generic LLMs still lack the capability to accurately interpret microscopic imagery at a level suitable for reliable use by biology scientists. Another tool that was investigated was Omega. Omega is a plug-in for Napari, an open-source image viewer widely used in scientific research, that integrates with existing LLM’s like ChatGPT for bioimage analysis. Omega is one of the first bioimage analysis tools created that can segment images, correct its own coding mistakes, and provide advice for processing images and analysis.4 However, because Omega is tightly coupled to Napari’s graphical user interface and does not support headless operation, it could not be run in a usable way on RON, a terminal-only Linux server. Early in the project, we identified DeepThought, a customizable language model platform developed by the UNH Research Computing Center (RCC) as a potential resource to support our work to try to automate the analysis of microscopic cell imagery. Deepthought supports Llama-3.2-Vision, a Multimodal Large Language Model (MLLM), that can process text as well as images. This multimodal capability was essential for our project, which involves the analysis of microscopic imagery to evaluate cell morphology. ￼

**Project Idea**

Our project, the ***Cell Morphology Analyzer***, leverages LLMs and computer vision techniques to identify morphological features indicative of cell death from fluorescent microscope images. The core idea is to integrate image preprocessing techniques with an LLM-based pipeline to analyze images, retrieve contextually relevant biological knowledge, and generate informed textual descriptions and classifications of observed morphological changes. The goal is to streamline the process of cell health evaluation, providing automated results with minimal human intervention, making it a valuable command line interface (CLI) tool for research in this area.

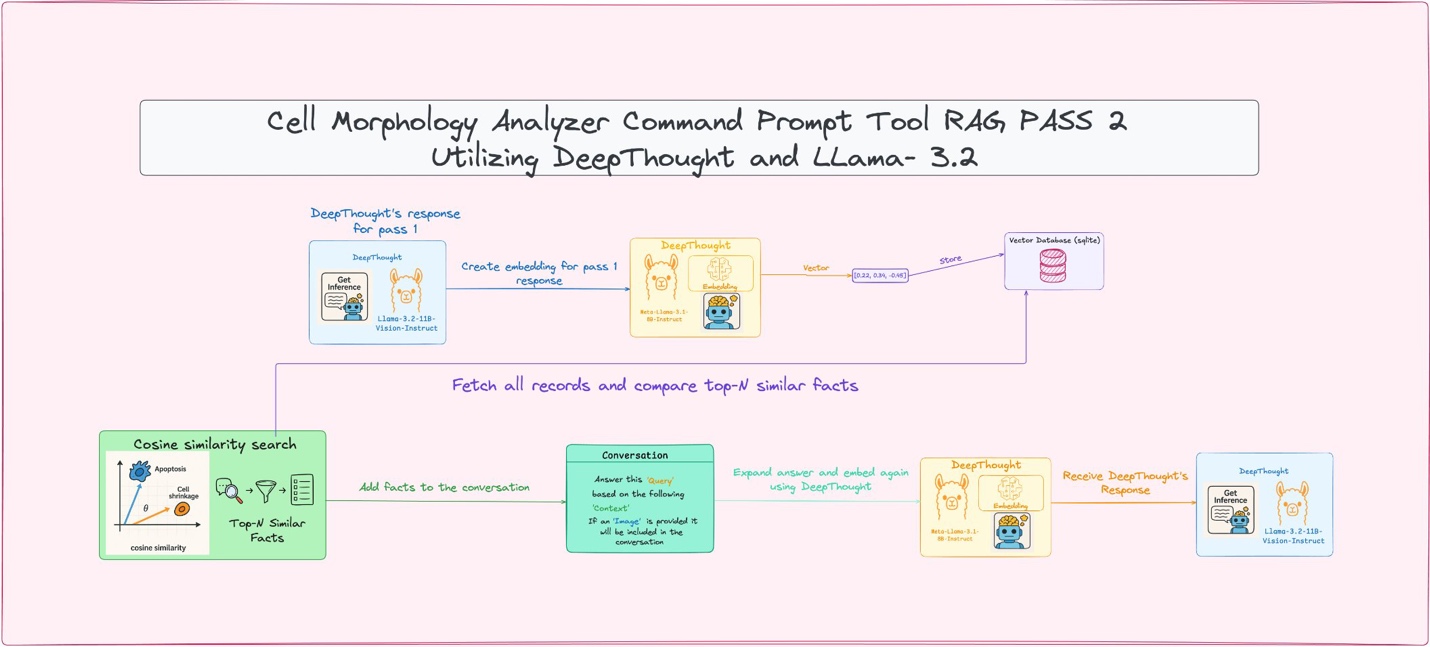
**Code Architecture and Design Decisions**

The **Cell Morphology Analyzer** is a modular software system developed in Python, designed to analyze cell morphology with the implementation geared to fluorescent microscopic imagery of cells. The system is structured to integrate several key technologies, including OpenCV for image processing, SQLite for storing biological data, and the DeepThought LLM for biological inference. The overall architecture is centered around a pipeline that processes microscopic images, retrieves relevant biological facts using the Retrieval-Augmented Generation (RAG) approach, and leverages LLama-3.2-Vision MLLM to generate insights into the cellular morphology. The code is divided into several independent components, each with a specific responsibility, ensuring flexibility, scalability, and ease of maintenance.

At a high level, the architecture consists of four core pieces: Image Preprocessing, RAG Database, DeepThought Integration, and the Analysis Pipeline. Image preprocessing is responsible for applying various transformations to microscopy images, including filtering, thresholding, edge detection, and segmentation. These transformations prepare the images for subsequent analysis by isolating critical cellular features while also reducing noise and artifacts. The DeepThought integration interfaces with the DeepThought LLM to generate answers based on both image data and a user query. We augment DeepThought’s performance with a Retrieval-Augmented Generation (RAG) mechanism using a two-pass dynamic knowledge injection technique. As seen in **Figure 1**, Pass 1 utilizes the user’s query and an image caption are embedded and used to retrieve relevant scientific context from a local SQLite RAG database that contains the vector embeddings of biological knowledge. In Pass 2, as seen in **Figure 2**, the initial LLM response is further refined by retrieving additional facts and prompting the model again for a more detailed or corrected output. This dual-pass system enables the model to reason more deeply about complex biological phenomena and adapt to nuanced morphological patterns in the image. These components work together to generate results for cell morphology image analysis. A key aspect of the system’s analysis pipeline is the parameter sweep and self-evaluation framework, which automatically tests and compares preprocessing configurations. This framework sweeps over various image processing parameters such as the *ksize* and *sigmaX* values for Gaussian blur, the thresholds for Canny edge detection, and the ranges for in-range thresholding. By iterating over the various configurations, the system self-evaluates the performance of each setup against a set of baseline cellular image-associated queries and ranks scores for each query and preprocessing configuration according to their accuracy. This automated evaluation ensures that the analysis pipeline is fine-tuned to deliver the best results for the microscopic imagery.

A diagram of a computer program

AI-generated content may be incorrect.Figure 1: Pass 1 of Cell Morphology Analyzer

Figure 2: Pass 2 of Cell Morphology Analyzer

**Image Preprocessing Pipeline**

Fluorescent microscopy images often contain noise and color channel complexity that can obscure critical morphological details. To address this, we developed a preprocessing pipeline using OpenCV and PIL that performs a sequence of operations designed to isolate and highlight structural features. These steps include Gaussian Blur to reduce noise, channel thresholding to isolate RGB components, Canny Edge Detection to identify contours, contour overlay to visualize structural outlines. This processed image is then passed to the LLM for further analysis. Users can select specific preprocessing configurations, such as applying only blurring or segmentation, enabling custom workflows depending on the experiment's needs. Initially, the system used the RGB color space to isolate the red, green, and blue channels separately. However, after conducting a parameter sweep to find the optimal preprocessing parameters, it was discovered that switching to the HSV color space significantly improved performance. HSV appears to better handle fluorescence intensities, which were essential for accurately distinguishing between cellular structures. This decision to move from RGB to HSV was one of the key optimizations in the image processing pipeline, leading to improved segmentation and feature extraction. Previous studies have shown that HSV outperforms RGB for cellular image segmentation, particularly in fluorescence microscopy, due to its ability to separate chromatic information from intensity, which is crucial for better handling the variations in fluorescence intensity across cellular structures.3 This shift in color space allowed for more precise identification and separation of features, particularly in images with varying fluorescent signal intensities. Later in the semester, came the use of *skimage.morphology.* This python library, part of scikit-image, provided a robust set of tools for cleanup of microscopic images. The *skimage.morphology* library provides useful morphological image operations like closing, hole filling, and object cleanup that allowed for the removal of small noise artifacts which ensured that only meaningful features were retained for further analysis. Closing, the dilation of an object followed by erosion, helped to fill in small gaps or holes that might have otherwise hindered accurate feature extraction, *binary\_fill\_holes* operation was used to close any gaps within the segmented regions. These morphological operations improved the quality of the segmented images, which was essential for the subsequent steps of the morphological analysis.

**Dynamic Retrieval-Augmented Generation (RAG)**

The Dynamic RAG system plays a pivotal role in refining the cell morphology analysis tool by integrating multiple passes of information retrieval and model inference. The process begins with the first pass, where the user’s query and an automatically generated caption of the image are embedded to identify the most relevant biological facts stored in the RAG database. These facts, which include details about fluorescence markers, apoptosis, and cell morphology, are retrieved based on their relevance to the query and the image’s content. The Cell Morphology Analyzer tool asks the Llama-3.2-Vison model via DeepThought to generate its own caption for the image. By having the LLM generate the caption itself using its own language and descriptions we hypothesized that it may improve the LLM’s understanding and model-training. In our evaluations we did see that the self-generated caption seemed to provide a richer context for the analysis and allowed the system to make more precise connections between the image and relevant biological facts. Studies have shown that LLM generated captions outperform user-written ones in terms of precision and recall across visual elements, leading to better performance in downstream tasks such as image-text retrieval and visual question answering.4 The RAG database stores the relevant biological facts in the form of their vector embeddings, which enables efficient fact retrieval. To retrieve the most relevant facts, cosine similarity is employed to compare the query embedding (which represents the user’s query and image data) with the stored embeddings in the database. Cosine similarity quantifies how closely aligned two vectors are in high-dimensional space, making it an effective measure of semantic similarity5. The database is queried by embedding both the user’s query and the image caption into vectors, which are then compared to the stored fact vectors using cosine similarity. This ensures that the facts that are retrieved are contextually relevant and accurate which enhances the analysis the LLM responds with. The multi-pass process is employed in the second pass to refine the initial response from the LLM. After generating the first response, the system generates an embedding for the response itself, and this embedding is compared against the RAG database to retrieve additional, more specific facts. These newly retrieved facts are used to refine the initial answer, ensuring the final response is more informed and detailed. This iterative process revisiting and augmenting the response through additional factual context is what makes the dynamic RAG approach particularly beneficial. The integration of the dynamic RAG database into the system ensures that the generated responses are rooted in relevant biological knowledge, thus improving the model’s reliability and precision in cellular image analysis.

**Analysis Pipeline**

The analysis pipeline is designed to evaluate the performance and accuracy of the Cell Morphology analyzer tool by running a parameter sweep and self-evaluation mechanism across a set of predefined conditions. Specifically, the system tests various image preprocessing configurations, including parameters such as Gaussian blur size, thresholding values, and edge detection settings. The evaluation process is conducted over five baseline questions related to cell morphology, using four fluorescent microscopy images of cells, against two prompt modes (dynamic rag or control) to assess the tool’s ability to identify morphological changes associated with different types of cells. Through this iterative testing, the pipeline automatically compares the performance of each configuration, ranks the results, and identifies the optimal preprocessing settings for the best analysis outcomes. This mechanism ensures that the system is fine-tuned to deliver the most accurate results for cellular image analysis.

**Experiment Evaluation/Code Information**

To ensure rigorous assessment, we implemented a parameter sweep framework designed to evaluate the performance of various preprocessing configurations. The preprocessing pipeline follows a fixed sequence: thresholding, blurring, Canny edge detection, segmentation, and contour drawing. What varies in our framework are the specific parameter configurations used within each preprocessing step. The parameters within each preprocessing step are systematically varied. For instance, we examine different kernel sizes and sigma values in Gaussian blur, edge detection thresholds in Canny, and contour drawing parameters such as method, color, and thickness. In the segmentation step, we apply color-based filtering using HSV (Hue, Saturation, Value) ranges. By adjusting the lower and upper bounds to capture various hues and saturations of interest in our case: blue for the nuclei, green for the lysosomes, and red for the mitochondria.

For each configuration, the analyzer processes the full image analysis pipeline across two distinct prompt modes: dynamic RAG and the control prompt mode. The dynamic RAG mode incorporates retrieval-augmented generation to enrich the model’s responses by retrieving contextually relevant facts from a knowledge base, while the control prompt simply generates a response from DeepThought without fact retrieval or external knowledge. This comparison allows for a more granular understanding of how dynamic RAG improves or alters performance when compared to the simpler control prompt mode. The system evaluates performance on five baseline questions applied to four fluorescent cell images, capturing the LLM’s generated response and comparing it to a ground truth table containing the correct answers for each question. This comparison enables DeepThought to self-assess its generated responses on a 1-5 scale. Additionally, our tool requires DeepThought to justify how and why it determines the score it assigns to its own performance. These results are stored in CSV files and visualized using Matplotlib facilitating a detailed comparative analysis. This analysis not only identifies the effectiveness of various preprocessing parameter settings but also highlights the performance differences between Dynamic RAG and the control prompt mode. Furthermore, the investigation explores how the number of relevant facts retrieved in dynamic RAG influences the accuracy and relevance of the model’s responses.

**Results and Discussion**

Preliminary results indicate that the integration of preprocessing with LLM analysis significantly improves the quality of morphological interpretation. The DeepThought model consistently demonstrated the ability to describe cell morphological changes such as membrane blebbing, nuclear fragmentation, and cytoplasmic shrinkage—hallmarks of apoptosis and other death pathways. Moreover, the LLM’s capacity for image-aware reasoning allowed it to generalize morphological cues across various cell types and staining methods. The two-pass RAG system enhanced contextual understanding, helping the model to refine its conclusions when initial outputs were vague or incorrect. In our evaluation, the parameter sweep framework was used to assess the performance of various preprocessing configurations applied to the image analysis pipeline. The preprocessing pipeline followed a fixed “default” sequence of thresholding, blurring, Canny edge detection, segmentation, and contour drawing, with specific parameters tested for each step. Among the configurations tested, the final preprocessing setup used a Gaussian blur with a kernel size of (5, 5), Canny thresholds of 50, and 150, and specific HSV ranges for segmentation which yielded the best results for the five baseline questions across the four fluorescent microscopic images.

This configuration likely performed best due to the balanced application of noise reduction, edge detection, and segmentation. The Gaussian blur with a kernel size of (5,5) and sigma X of 0 effectively smoothed out noise while preserving important cellular features, ensuring that subsequent edge detection with the Canny algorithm could more accurately identify boundaries and structures. The Canny thresholds of 50 and 150 were well-calibrated to detect edges in the images without introducing too much noise, which is crucial when analyzing complex fluorescent staining patterns.

One of the most impactful components of this preprocessing pipeline was the color-based segmentation using HSV (Hue, Saturation, Value) ranges. In particular, the specific values chosen for the OpenCV *in\_range* function were critical for accurately segmenting the different organelles. The HSV range for blue (for the nuclei) was set as lower blue of: (100, 50, 50) and an upper blue of: (150, 255, 255). These values effectively captured the typical hue and saturation of the DAPI stain used for labeling the nucleus while avoiding unnecessary noise from other parts of the image. The choice of the lower blue and upper blue values from the blue channel was crucial in isolating the nucleus while keeping out background colors or weak fluorescent signals from nearby structures. The lower bound ensured that only the blue shades associated with DAPI staining were included, while the upper bound allowed for sufficiently wide range to capture different intensities of the blue stain. For green (lysosomes), the HSV values were set as lower green of:(40, 50, 20) and upper green of: (80, 255, 255). The values captured the characteristic green fluorescent of the GFP stain used for labeling lysosomes. For the red channel, which corresponds to mitochondria, the first HSV range for red was set to a lower red of: (0, 100, 100) and an upper red of: (10, 255, 255). The second HSV range for the red was set to a lower red of: (160, 100, 100) and an upper red of: (180, 255, 255). In the HSV color space, red spans two distinct regions: one at the low end of the hue spectrum (near 0°) and another at the high end (near 360°). This dual occurrence of the red in the hue spectrum is why we used two separate ranges in the pipeline for red. By using these two ranges, we ensure that both weaker and more intense red fluorescence signals, typical of rhodamine-stained mitochondria, are captured effectively.

When determining the best preprocessing configuration, we observed that certain HSV range values did not perform as well due to their broader range spans. For instance, the range for red organelles used a span for lower red and upper red that was too wide in one configuration set which led to the inclusion of unwanted background signals. Similarly, certain configurations for the blue and green ranges were not sufficiently restrictive, causing the system to either miss relevant parts of the image or pick up unnecessary noise from unrelated regions. These findings highlight the importance of fine-tuning the HSV ranges to avoid both excessive inclusion of irrelevant pixels and overly restrictive bounds that may fail to capture important features. The contour drawing step, with parameters set to draw all contours in green at a thickness of 2 rather than 1 further emphasized the key features, helping the system to more clearly delineate the organelles of interest and assess any morphological changes. This effective segmentation of key structures contributed to the improvement in performance across most queries.

The dynamic RAG and control prompt modes performed similarly overall, each achieving an accuracy of 63.5%. The evaluation of our cell morphology analyzer tool revealed that “default” preprocessing generally improved performance over no preprocessing, particularly in dynamic RAG mode. The final accuracy for dynamic RAG and control modes was 63.5% across all queries. Default preprocessing outperformed no preprocessing by 3 percentage points (65% vs 62%), with preprocessing contributing to improved accuracy in queries related to morphological changes and fluorescence analysis.

For the query *“What are the fluorescent dyes being used in this image if any?”*, dynamic RAG with “default” preprocessing achieved 65% accuracy, compared to 60% accuracy with no preprocessing. The query “*Do the cells in the images seem healthy?”* showed a stronger result for the control prompt more (80% with “default” vs 65% for dynamic RAG), indicating that preprocessing was beneficial overall, but not when it came to a more open-ended cell health query. The query *“Do you notice any morphological changes?”* demonstrated a clear advantage for “default” preprocessing, with dynamic RAG achieving 75% accuracy, compared to 70% with no preprocessing. The query related to cell death estimation, however, remained challenging for our tool with dynamic RAG in default preprocessing achieving 55% accuracy, which was significantly higher than the 40% seen in the control prompt mode.

These results show that while preprocessing clearly improves performance, further refinements are needed, particularly in better differentiating cells states and improving the accuracy of cell death predictions. The detailed comparison, including “default” preprocessing performance versus no preprocessing, is illustrated in **Figure 3,** which shows the 1-5 scores for each query and each image for dynamic RAG and control prompt modes with and without preprocessing applied to the images.

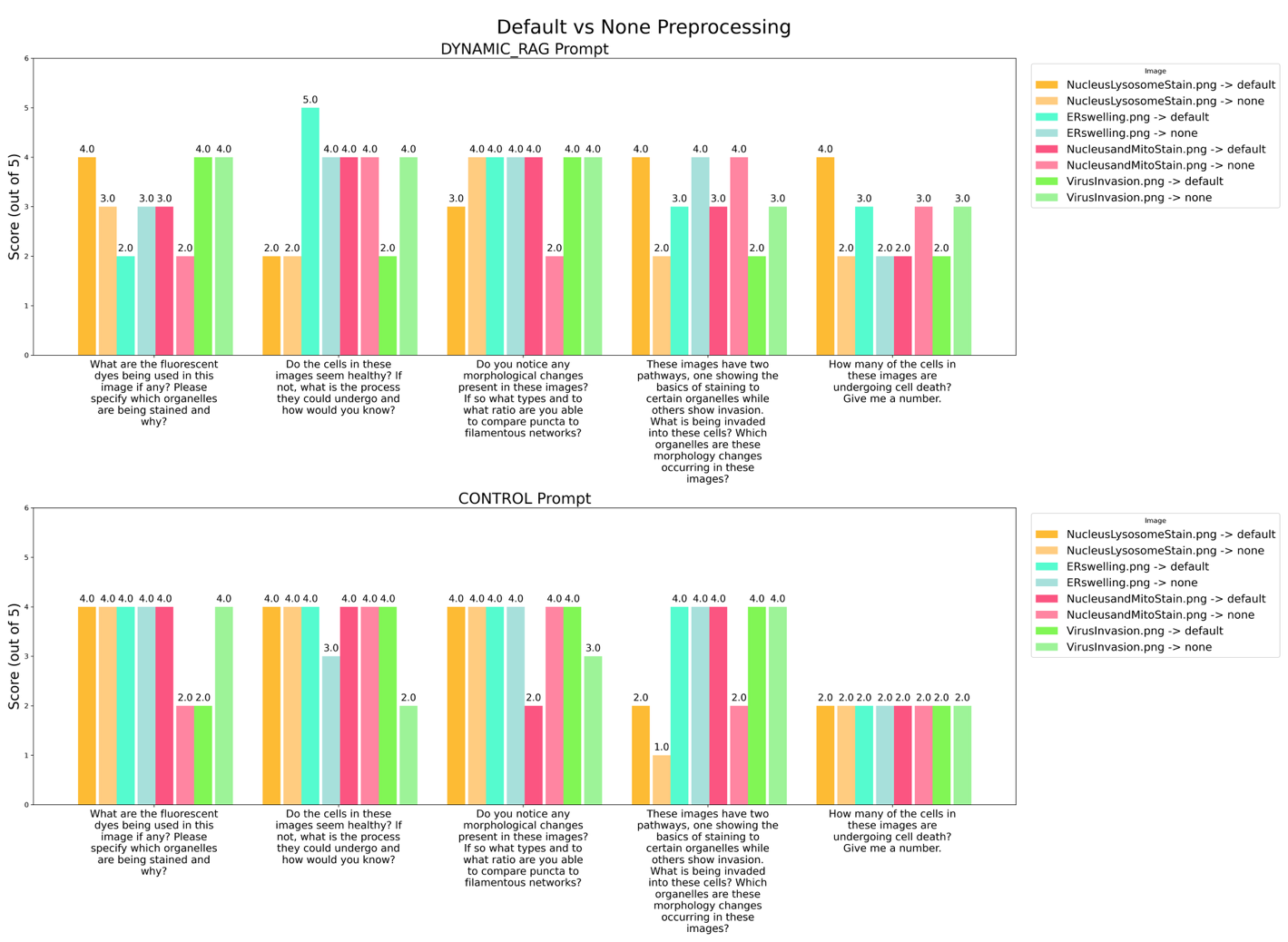


Figure 3: Results from LLM showing evaluations of its own responses to 5 questions on a 1-5 scale for 4 images.

**Future Directions**

Several extensions are planned to further enhance the Cell Morphology Analyzer via the Dynamic RAG Training by incorporating fine-tuned embeddings based on biomedical literature to improve retrieval precision. In future development it may be useful to develop a graphical user interface for use on the web or local desktop application for broader accessibility to life science researchers. There is also room to improve the scalability of the pipeline for batch analysis of image datasets from time-lapse microscopy. Lastly, automated labeling will allow for training the model to generate high-quality labeled datasets to support downstream supervised learning. If time and resources permitted, further analysis of the preprocessing parameter sweep could be conducted to refine the image quality fed into DeepThought, potentially improving its performance and resulting self-assessment scores across the 5 baseline questions. Ultimately, the goal is to make LLMs not just interpreters, but active collaborators in biological discovery—bridging visual data with textual scientific understanding in real time.

**Conclusion**

Our project demonstrates the feasibility and utility of using large language models to automate and enhance the interpretation of fluorescent microscopy images. By combining sophisticated preprocessing with RAG-enabled LLMs, we have created a powerful tool capable of identifying morphological changes associated with cell death. This represents a significant step forward in the application of AI to life sciences, offering scalable, reproducible, and insightful image analysis for biological research.

**Citations**

(1)

*10 Benefits and 10 Challenges of Applying Large Language Models to DoD Software Acquisition*. <https://insights.sei.cmu.edu/blog/10-benefits-and-10-challenges-of-applying-large-language-models-to-dod-software-acquisition/> (accessed 2025-05-01).

(2)

Cooper, G. M. The Origin and Evolution of Cells. In *The Cell: A Molecular Approach. 2nd edition*; Sinauer Associates, 2000.

(3)

ElDahshan, K. A.; Youssef, M. I.; Masameer, E. H.; Mustafa, M. A. Comparison of Segmentation Framework on Digital Microscope Images for Acute Lymphoblastic Leukemia Diagnosis using RGB and HSV Color Spaces. *J. Biomed. Eng. Med. Imaging* **2015**, *2* (2), 26–34. <https://doi.org/10.14738/jbemi.22.1065>.

(4)

Royer, L. A. Omega — Harnessing the Power of Large Language Models for Bioimage Analysis. *Nat Methods* **2024**, *21* (8), 1371–1373. <https://doi.org/10.1038/s41592-024-02310-w>.

(5)

You, K. Semantics at an Angle: When Cosine Similarity Works Until It Doesn't. arXiv 2025. <https://arxiv.org/abs/2504.16318>.[arXiv](https://arxiv.org/abs/2504.16318?utm_source=chatgpt.com)

(6)

Zhang, J.; Li, Z.; Wu, Y.; Liu, Y.; Li, Z. Benchmarking and Improving Detailed Image Captioning. *arXiv* 2023, <https://arxiv.org/abs/2405.19092v2>.