A Strategic Plan for the Computational Analysis of *Candida albicans* Biofilm Dynamics

Section 1: Strategic Overview and Phased Execution Plan

This document presents a comprehensive technical and strategic plan for the development of a computational pipeline to analyze time-lapse microscopy images of *Candida albicans*. The primary objective is to identify, quantify, and characterize the dynamic processes of biofilm formation and yeast-form cell dispersal. The plan is structured to systematically address the project's evaluation criteria, from the "Minimal Model" requirements to the more complex "Icing on the Cake" objectives. It outlines a phased execution model aligned with the competition schedule, emphasizing a robust, reproducible, and scalable workflow built upon modern machine learning operations (MLOps) principles. The strategic foundation of this plan is a modular architecture, where specialized computational modules are developed to answer specific biological questions posed by the research context, transforming the final system from a mere predictive tool into a quantitative instrument for mycological research.

1.1 The Central Challenge: Overcoming the Low-Magnification Barrier

The single most significant technical challenge presented by this project is the low magnification (10x) of the source microscopy data. In optical microscopy, magnification and resolution are distinct concepts; while magnification enlarges the image, resolution determines the ability to distinguish fine details. At low magnifications, the spatial resolution is fundamentally limited by the diffraction of light, meaning that features smaller than a certain threshold will be blurred or lost entirely, a phenomenon known as "empty magnification". The research context specifies the need to distinguish between different subpopulations of dispersed cells, such as round and oval phenotypes. At 10x magnification, these subtle morphological differences may be represented by only a handful of pixels, making them exceptionally difficult to resolve and classify using standard computer vision algorithms. This low resolution degrades object edge detail and reduces the signal-to-noise ratio, complicating segmentation and detection tasks.
To surmount this critical obstacle, the core technical strategy of this project will be an **"Enhance-then-Analyze"** approach. This strategy posits that a dedicated pre-processing step to computationally increase the effective resolution of the images is not an optional enhancement but a foundational necessity for success. The project will therefore incorporate the development and application of a Single Image Super-Resolution (SISR) model as the first step in the analytical pipeline. By training a deep learning model, such as a Generative Adversarial Network (GAN), to learn the mapping from low-resolution to high-resolution images, it is possible to reconstruct finer details and sharper edges that were latent in the original data. This technique has proven highly effective in various domains, including medical and microscopic imaging, where it enhances the performance of subsequent analysis tasks. Applying a

super-resolution prior will effectively transform the low-quality input into a high-quality substrate suitable for the demanding segmentation, characterization, and tracking models required to achieve the project's most ambitious goals.

1.2 A Phased, Iterative Development Model

To ensure a structured and risk-managed execution, the project will be divided into a four-phase iterative development model. This structure is deliberately designed to align with the principles of agile development, prioritizing the delivery of a functional baseline system early in the project lifecycle. By focusing on the "Minimal Model" requirements first, the project secures a solid foundation and mitigates the risk of failing to meet core deliverables while pursuing more complex, high-risk objectives. Each phase builds upon the last, progressively increasing the sophistication and analytical richness of the computational pipeline.

- **Phase I (Weeks 1-4): Foundational Analysis, Environment Setup, and Baseline Modeling.** This initial phase is dedicated to risk reduction and establishing a professional workflow. It involves a deep exploratory data analysis (EDA) to fully characterize the dataset, the setup of a scalable and reproducible MLOps environment on a chosen cloud platform, and the development of simple, classical computer vision models to establish performance baselines. The outputs of this phase will directly inform the "Exploratory Data Analysis Slides" deliverable.
- **Phase II (Weeks 5-9): Core Deep Learning Pipeline for Minimal Requirements.** This phase focuses on developing the high-performance deep learning models necessary to meet the three core "Minimal Model" criteria. This includes a U-Net model for biofilm segmentation, a super-resolution and object detection pipeline for dispersed cell quantification, and a time-series analysis module for identifying the initiation of dispersal.
- **Phase III (Weeks 10-13): Advanced Modeling for Characterization and Tracking.** Building on the outputs of the core pipeline, this phase tackles the "Icing on the Cake" objectives. This involves upgrading the detection model to an instance segmentation model for detailed cell characterization, implementing skeletonization and graph analysis for biofilm characterization, and developing a sophisticated multi-object tracking system to analyze post-dispersal cell dynamics.
- **Phase IV (Weeks 14-15): System Integration, Final Evaluation, and Reporting.** The final phase is dedicated to integrating all developed modules into a single, automated end-to-end pipeline. It also includes large-scale hyperparameter tuning, final performance evaluation against the specified criteria, and the preparation of all submission materials, including the final report, presentation, and codebase.

1.3 Project Roadmap and Milestone Alignment

The successful execution of this project requires meticulous alignment of the development phases with the competition's schedule. The following roadmap integrates the four-phase model with the key dates and deliverables provided, ensuring that all deadlines are met and that sufficient time is allocated for each critical task. This schedule serves as the master plan, providing clarity on dependencies and priorities throughout the project lifecycle.

Task/Milestone	Phase	Primary Sprint(s)	Start Date	End Date	Key Deliverable(s)	Corresponding Competition Deadline
Project Kickoff & Team Registration	I	1	9/11	9/18	Registered Team, Project Charter	9/18 (Team Registration)
Exploratory Data Analysis (EDA) & MLOps Setup	I	1	9/12	9/26	EDA Notebook, Cloud Environment Setup	N/A
Baseline Modeling & EDA Presentation Prep	I	2	9/25	9/30	Baseline Model Code, EDA Slides	9/30 (EDA Slides Due)
EDA Presentations	I	2	10/2	10/2	Presented Slides	10/2 (EDA Presentations)
Super-Resolution & U-Net Biofilm Segmentation	II	2, 3	9/25	10/17	Trained SR Model, Trained U-Net Model	N/A
Dispersed Cell Detection (YOLO) & Quantification	II	3, 4	10/9	10/31	Trained Detector, Cell Count Time Series	N/A
Change Point Detection for Dispersal Onset	II	4	10/23	11/7	Dispersal Onset Prediction Script	N/A
Progress Report Preparation	II	4	11/1	11/11	Progress Report Slides	11/11 (Progress Report Slides Due)
Draft Solution Presentatio	II	5	11/13	11/13	Presented Draft Solution	11/13 (Draft Solution Presentation

Task/Milestone	Phase	Primary Sprint(s)	Start Date	End Date	Key Deliverable(s)	Corresponding Competition Deadlines)
Instance Segmentation for Cell Characterization	III	5	11/14	11/22	Trained Instance Segmentation Model	N/A
Morphological Feature Extraction (Cells & Biofilm)	III	5, 6	11/14	11/29	Feature Extraction Pipeline	N/A
Multi-Object Tracking (DeepSORT)	III	6	11/21	12/5	Trained Appearance Model, Tracking Module	N/A
End-to-End Pipeline Integration & Final Tuning	IV	6	12/4	12/6	Automated Pipeline Script	N/A
Final Project Submissions	IV	6	12/7	12/7	Final Codebase, Report, Video	12/7 (Final Project Submissions Due)
Final Slides Preparation	IV	6	12/8	12/9	Final Presentation Slides	12/9 (Final Slides Due)
Final Presentations	IV	6	12/11	12/11	Final Presentation	12/11 (Final Presentations)

1.4 Strategic Implications of the Project Structure

The structure of this project plan is predicated on the understanding that the task is not to build a single, monolithic machine learning model. Instead, it is to construct a multi-stage computational pipeline that mirrors a scientific investigation. The evaluation criteria are deliberately partitioned into distinct facets of image analysis—segmentation, detection, time-series analysis, and tracking—which directly correspond to the biological questions outlined in the provided research context. For instance, generating a biofilm growth curve computationally addresses the biological question of biofilm expansion dynamics. Identifying the initiation of dispersal with a statistical model provides a quantitative answer to when this key lifecycle stage begins. Characterizing dispersed cells computationally is a direct proxy for the laboratory assays described in the research aims.

This correspondence between computational tasks and biological questions necessitates a modular project architecture. Each component of the pipeline—the segmentation module, the detection module, the tracking module—can be developed, tested, and validated independently before being integrated into the final workflow. This modularity confers significant advantages: it allows for parallel development by team members, simplifies debugging by isolating potential points of failure, and enables the substitution of one algorithmic approach for another (e.g., swapping a YOLO detector for a Mask R-CNN) without requiring a complete system overhaul. Ultimately, the success of this project will be measured not only by the accuracy of its predictions but also by its utility as a scientific instrument for quantitative biology. The final report and presentation should therefore frame the results within this narrative. For example, a statement such as, "The growth curve generated by our U-Net segmentation pipeline quantitatively demonstrates the cessation of biofilm expansion, which our change point detection model identifies as coinciding with the onset of dispersal, thereby providing computational validation for the hypothesis of a distinct, regulated dispersal phase" elevates the project from a technical exercise to a meaningful contribution to the understanding of *C. albicans* biology. This framing demonstrates a deeper level of comprehension and is likely to be rewarded in the final evaluation.

Section 2: Phase I - Foundational Analysis and Workflow Establishment (Weeks 1-4)

The initial phase of the project is paramount for establishing the technical and operational groundwork necessary for success. Its primary goals are to de-risk subsequent development phases by thoroughly understanding the data, to build a robust and reproducible development environment that will prevent common collaborative pitfalls, and to create simple baseline models that will both inform the "Exploratory Data Analysis Slides" and serve as a crucial benchmark against which more sophisticated models will be measured. This phase lays the foundation for a professional, efficient, and scalable project execution.

2.1 Deep Dive: Exploratory Data Analysis (EDA) for Microscopy Time-Lapse

Before any model development, a systematic characterization of the visual properties of the dataset is essential. This exploratory data analysis will identify potential challenges and inform the design of pre-processing steps and model architectures.

- **Image Quality Assessment:** A representative subset of images from different time points and experiments will be analyzed to quantify fundamental properties. This includes generating histograms of pixel intensities to assess brightness and contrast, and calculating signal-to-noise ratios to understand the clarity of the cellular structures against the background. This process will also involve a meticulous visual inspection to identify and document common microscopy artifacts. These may include uneven illumination (vignetting), where the image periphery is darker than the center; out-of-focus debris or contaminants on the slide; and potential focus drift over the 20-hour acquisition period, which could render certain time points unusable or require special handling.
- **Temporal Consistency Analysis:** Time-lapse microscopy is susceptible to temporal artifacts that can confound analysis. For instance, the output of microscope lamps can

fluctuate as they warm up or age over long experiments. To detect such issues, the mean and standard deviation of pixel intensity for each full frame will be calculated and plotted against time. Any significant, systematic trends in these metrics could indicate an instrumental artifact rather than a biological change, and would necessitate a temporal normalization step in the pre-processing pipeline to avoid biasing intensity-based segmentation or detection algorithms.

- **Visual Morphology Dictionary:** To bridge the gap between the biological description and the low-resolution image data, a visual dictionary will be created. This involves manually inspecting and annotating a small but diverse set of frames to document the characteristic appearance of key biological structures as they appear at 10x magnification. This dictionary will include examples of: nascent biofilms (small clusters of yeast cells), mature hyphal networks (interwoven filamentous structures), individual round yeast-form cells, individual elongated yeast-form cells, and clumps of adherent dispersed cells. This qualitative analysis is crucial for establishing a visual ground truth, informing the annotation process for model training, and providing a realistic assessment of the difficulty of each analytical task.

2.2 MLOps Foundation: A Reproducible and Scalable Environment

To prevent the project from descending into disorganized, irreproducible experimentation, a professional-grade development environment based on MLOps principles will be established from the outset. The scheduled workshops on AWS SageMaker, IBM Watsonx.ai, and GCP Vertex AI provide an ideal opportunity to select and implement a primary cloud platform for this purpose. GCP Vertex AI, for example, offers a unified platform that integrates the necessary tools for a streamlined workflow.

The core components of this MLOps foundation will include:

- **Centralized Data Storage:** All raw, annotated, and processed datasets will be stored in a centralized cloud storage solution, such as Google Cloud Storage or Amazon S3. This ensures that all team members are working from a single, versioned source of truth, eliminating data-related inconsistencies.
- **Version Control:** A Git repository will be initialized to manage all project assets, including source code, Jupyter notebooks, model configuration files, and documentation. This is fundamental for tracking changes, collaborating on code, and ensuring the ability to revert to previous versions if necessary.
- **Collaborative Development Environment:** A managed notebook service, such as Vertex AI Workbench or Amazon SageMaker Studio, will be used for all collaborative development. These platforms provide pre-configured environments with necessary libraries, ensuring that all team members have identical software dependencies. They also offer seamless integration with data storage and compute resources, facilitating efficient experimentation.
- **Systematic Experiment Tracking:** From the very first model run, an experiment tracking tool will be integrated into the workflow. Services like MLflow (available in SageMaker) or Vertex AI Experiments allow for the automatic logging of every experiment, including the code version, model parameters, performance metrics, and output artifacts. This practice is non-negotiable for ensuring reproducibility, enabling systematic comparison of different model architectures and hyperparameters, and maintaining a clear audit trail of the model development process.

2.3 Baseline Modeling: Simple Methods for Initial Quantification

To establish a performance baseline and generate initial results for the EDA presentation, simple, classical computer vision models will be implemented for the "Minimal Model" criteria. These baselines are crucial for two reasons: they provide a quick, first-order approximation of the biological dynamics, and they create a quantitative benchmark that justifies the subsequent use of more computationally expensive and complex deep learning models.

- **Biofilm Growth Curve (Baseline):** An adaptive thresholding method, such as Otsu's algorithm, will be applied to each frame to binarize the image into biofilm and background pixels. Morphological operations, such as morphological opening (erosion followed by dilation) and closing (dilation followed by erosion), will be used to remove noise and fill in small holes within the biofilm mask. The total pixel area of the resulting biofilm mask will be calculated for each frame and plotted over time to generate a preliminary growth curve.
- **Dispersed Cell Quantification (Baseline):** For detecting small, round objects like dispersed yeast cells, a Laplacian of Gaussian (LoG) filter will be applied. This filter is effective at highlighting blob-like structures of a specific size. By finding the local maxima in the filtered image, it is possible to identify and count putative cell locations. While this method is expected to be noisy and prone to false positives, it will provide a valuable initial estimate of the dispersed cell count over time.

The plots and methods developed in this step will form the quantitative core of the EDA presentation, demonstrating a proactive, data-driven approach and a clear understanding of the project's core challenges.

2.4 Strategic Implications of the MLOps Foundation

The scheduled workshops on cloud ML platforms should be viewed not as passive learning sessions, but as strategic opportunities to build a scalable and governable MLOps foundation for the project. The challenges of a multi-month, team-based ML project—managing multiple datasets, tracking hundreds of experiments, and ensuring final models can be reliably reproduced—are precisely the problems that platforms like SageMaker, Watsonx, and Vertex AI are designed to solve. Failing to adopt such a structured approach from the beginning introduces significant project risk, often leading to a chaotic final phase characterized by inconsistent results, lost experimental data, and an inability to reproduce the best-performing model.

Therefore, the team's strategy should be to actively implement the project *within* the chosen MLOps framework immediately following the introductory workshops. For example, after the Vertex AI introduction, the team should create a Vertex AI Project, connect it to the Git repository, configure Vertex AI Experiments for tracking, and begin running EDA notebooks within the managed Vertex AI Workbench environment. This proactive adoption of MLOps principles provides a significant competitive advantage. While other teams may struggle with integration and reproducibility in the final weeks, this project's pipeline will be robust, versioned, and easily demonstrable. This high degree of professionalism and technical rigor will be evident in the final submission, reflecting a mature approach to machine learning engineering that is likely to be recognized and rewarded in the evaluation.

Section 3: Phase II - Core Pipeline Development for Minimal Requirements (Weeks 5-9)

With the foundational analysis complete and a robust MLOps environment established, Phase II focuses on the development of a high-performance deep learning pipeline engineered to meet and exceed the three "Minimal Model" evaluation criteria. This phase will transition from classical computer vision baselines to state-of-the-art deep learning architectures, culminating in a working system capable of generating accurate, quantitative outputs for biofilm growth, dispersed cell counts, and the precise timing of dispersal initiation.

3.1 Task 1: Biofilm Growth Curve via Semantic Segmentation

The objective of this task is to produce a highly accurate growth curve by precisely segmenting the total biofilm area—encompassing both yeast-form and hyphal cells—from the background in every frame of the time-lapse.

- **Methodology: U-Net Architecture.** The chosen architecture for this task is the U-Net, a convolutional neural network specifically developed for biomedical image segmentation. Its selection is justified by several key advantages. The U-Net's symmetric encoder-decoder structure, featuring "skip connections" that concatenate feature maps from the contracting path to the expansive path, allows it to capture both high-level context and precise low-level localization details. This is critical for accurately delineating the complex and fine boundaries of hyphal filaments. Furthermore, U-Net has demonstrated exceptional performance even when trained on a relatively small number of annotated images, a common constraint in biomedical applications, making it perfectly suited for this project.
- **Implementation Plan:**
    1. **Data Preparation:** The provided annotations will be used to generate binary segmentation masks, where pixels belonging to the biofilm are labeled with 1 and background pixels are labeled with 0.
    2. **Data Augmentation:** To train a robust model that generalizes well, aggressive data augmentation will be applied. This is a critical step to artificially expand the training dataset and teach the model invariance to variations not present in the limited annotated set. Augmentations will include geometric transformations relevant to microscopy, such as random rotations, horizontal and vertical flips, and elastic deformations, as well as photometric adjustments like random changes in brightness and contrast.
    3. **Model Training:** The U-Net model will be trained using a composite loss function, typically a weighted sum of Binary Cross-Entropy (BCE) loss and Dice loss. BCE loss evaluates the classification error for each pixel independently, while Dice loss is better at handling class imbalance (often more background than foreground) and encourages the model to produce masks with high overlap with the ground truth. The entire training process, including hyperparameters and performance metrics (e.g., validation Dice coefficient), will be meticulously logged using the established MLOps platform.
    4. **Post-processing:** The raw output masks from the trained model may contain small, spurious predictions. Morphological operations, such as removing all

connected components below a certain size threshold, will be applied as a final clean-up step to improve the quality of the segmentation.

5. **Output Generation:** The trained and validated pipeline will be run on the full time-lapse sequence. For each frame, the total area of the final, post-processed biofilm mask will be calculated. Using the image's pixel-to-micrometer scale factor, this area will be converted to square micrometers (μm^2) and plotted against time to generate the final, high-precision growth curve.

3.2 Task 2: Dispersed Cell Quantification via Object Detection

The objective of this task is to accurately detect and count the number of individual dispersed yeast-form cells in each frame of the time-lapse. This task directly implements the "Enhance-then-Analyze" strategy.

- **Sub-Task 2.1: Super-Resolution Pre-processing.** As established, the native 10x magnification is insufficient for reliably detecting small cellular objects. Therefore, the first step will be to apply a super-resolution model to the raw images. A model based on Generative Adversarial Networks (GANs), such as an Enhanced Super-Resolution GAN (ESRGAN), will be trained to upscale the images by a factor of 2x or 4x. This process computationally generates a high-resolution version of each frame, enhancing sharpness and recovering fine details that are critical for the subsequent detection model to perform accurately.
- **Methodology: YOLO (You Only Look Once) Object Detection.** For the detection and counting task, a model from the YOLO family (e.g., YOLOv5 or YOLOv8) will be employed. YOLO models are state-of-the-art, single-stage detectors that offer an exceptional balance of speed and accuracy, making them highly suitable for high-throughput analysis of microscopy images. For the minimal requirement of simply counting the cells, an object detection approach (which outputs bounding boxes) is more efficient and straightforward than instance segmentation (which outputs pixel masks).
- **Implementation Plan:**
    1. **Input Data:** The input to the YOLO model will be the high-resolution images generated by the super-resolution model in the previous sub-task.
    2. **Model Training:** A YOLO model pre-trained on a large-scale dataset (like COCO) will be fine-tuned using the provided annotations of dispersed cells. Transfer learning in this manner allows the model to leverage general feature extraction capabilities and adapt them to the specific task of cell detection with a smaller, domain-specific dataset.
    3. **Output Generation:** When applied to a frame, the trained YOLO model will output a list of bounding boxes, each corresponding to a detected dispersed cell. The final count for that frame is simply the total number of predicted bounding boxes. This process is repeated for every frame to generate a time series of dispersed cell counts.

3.3 Task 3: Identifying Dispersal Initiation via Change Point Detection

The objective here is to pinpoint the specific frame (and thus, the time point) at which the biological process of dispersal begins. This is fundamentally a time-series analysis problem, not a direct image analysis task.

- **Methodology: Statistical Change Point Detection.** The time series of dispersed cell

counts generated in Task 3.2 provides the necessary input. The onset of dispersal will manifest as an abrupt change in the statistical properties of this time series—specifically, a shift in the mean count from a value near zero to a sustained, positive, and likely increasing value. Change Point Detection (CPD) is a class of statistical algorithms explicitly designed to identify such abrupt transitions in time-series data.

- **Implementation Plan:**
    1. **Input Data:** The input will be a one-dimensional vector, C = [c_1, c_2,..., c_n], where c_t is the dispersed cell count for frame t as determined by the YOLO pipeline.
    2. **Algorithm Selection:** An offline CPD algorithm will be used, as the entire time series is available for analysis. A suitable choice is the Pruned Exact Linear Time (PELT) method, which is computationally efficient and guaranteed to find the optimal segmentation of a time series based on a chosen cost function (e.g., change in mean). The algorithm will be configured to detect changes in the mean of the time series and to identify the *first* statistically significant change point.
    3. **Output Generation:** The CPD algorithm will return the index of the frame that marks the beginning of the new statistical regime, which directly corresponds to the identified onset of dispersal. This approach provides a quantitative, reproducible, and statistically defensible answer to this evaluation criterion.

3.4 Strategic Implications of the Core Pipeline's Dependencies

A critical realization for this phase is that the three "Minimal Model" tasks are not independent but form a tightly coupled, dependent pipeline. The accuracy of the final task, dispersal initiation detection, is entirely contingent on the quality of the outputs from the preceding tasks. This creates a clear hierarchy of dependencies: the performance of the Change Point Detection algorithm (Task 3.3) depends directly on the accuracy and stability of the cell counts from the YOLO detector (Task 3.2). In turn, the YOLO detector's performance is profoundly influenced by the effectiveness of the super-resolution pre-processing step (Task 3.2.1).
This cascading dependency has significant strategic implications for resource allocation, debugging, and optimization. If the final predicted dispersal time is inaccurate, the root cause is unlikely to be a flaw in the CPD algorithm itself, which is a standard statistical method. The investigation must proceed upstream. The first step would be to visually inspect the output of the YOLO detector on frames around the supposed dispersal time. Are cells being missed? Are there false positives in early frames creating a noisy baseline? If the detector's performance is suboptimal, the next step is to analyze its input: the super-resolved images. Are they blurry, or do they contain artifacts? Is the super-resolution model failing to enhance the relevant cellular features? This establishes a clear debugging hierarchy: **Super-Resolution → Detection → Time-Series Analysis**. Understanding this dependency structure is crucial for efficient problem-solving, preventing the team from wasting valuable time attempting to fine-tune the CPD algorithm when the true error lies in the upstream image processing and detection stages. This strategic understanding is formalized in the following table, which justifies the key algorithmic choices for the entire project.

Task	Chosen Algorithm	Rationale/Justification	Alternative(s) Considered	Reason for Rejection
Biofilm Segmentation	U-Net	State-of-the-art for biomedical	Mask R-CNN	More complex, designed for

Task	Chosen Algorithm	Rationale/Justification	Alternative(s) Considered	Reason for Rejection
		segmentation; excellent performance with limited data; skip connections preserve fine details crucial for hyphae.		instance segmentation, which is overkill for measuring total biofilm area. Higher computational cost for training and inference.
Dispersed Cell Detection	Super-Resolution (ESRGAN) + YOLOv8	The "Enhance-then-Analyze" strategy is essential to overcome 10x magnification. YOLO provides a fast and accurate bounding-box detector for counting.	Classical Blob Detection (LoG)	Used as a baseline, but highly susceptible to noise and lacks the robustness of a deep learning approach, leading to inaccurate counts.
Dispersal Onset Detection	Change Point Detection (PELT)	The correct statistical tool for identifying abrupt shifts in a time series (cell counts). It is quantitative, reproducible, and robust.	Manual/Heuristic Thresholding	Subjective, not reproducible, and likely to fail with noisy count data. Lacks statistical rigor.
Cell Tracking	DeepSORT	Incorporates a deep appearance model, making it robust to occlusions and re-identification errors, which are common in crowded cell cultures. Outperforms motion-only trackers like SORT.	SORT	Relies only on IoU and motion, leading to frequent ID switches during occlusions, which would corrupt post-dispersal growth analysis.

Section 4: Phase III - Advanced Modeling for "Icing on

the Cake"

With the minimal requirements secured by the core pipeline, Phase III is dedicated to developing the advanced analytical modules required to address the "Icing on the Cake" evaluation criteria. This phase leverages the outputs of the core pipeline—such as segmentation masks and initial detections—to extract a richer layer of biological information. The focus shifts from simple quantification to detailed characterization of both individual cells and the collective biofilm structure, as well as tracking cellular dynamics over time.

4.1 Dispersed Cell Characterization: From Bounding Boxes to Phenotypes

The objective of this task is to move beyond simply counting dispersed cells to characterizing their individual morphological properties, specifically their size and shape (elongation), as well as identifying cell clumps. This directly addresses the research goal of identifying and analyzing subpopulations of dispersed cells.

- **Methodology: Instance Segmentation and Feature Extraction.** To perform accurate morphological analysis, a simple bounding box is insufficient. A pixel-perfect mask for each individual cell is required.
    - **Refining Detection:** The YOLO object detector developed in Phase II will be replaced or supplemented with a state-of-the-art instance segmentation model, such as Mask R-CNN or the segmentation-capable version of YOLOv8 (YOLOv8-Seg). These models, when applied to an image, output three things for each detected object: a class label, a bounding box, and a high-resolution pixel mask. This model will also be trained on the super-resolved images to ensure it can capture fine cell boundaries.
    - **Feature Extraction Pipeline:** A post-processing pipeline will be developed to analyze the instance masks generated by the segmentation model. For each individual cell mask, the following morphological features will be calculated using established image analysis libraries like scikit-image or OpenCV :
        1. **Cell Size:** This is calculated as the total number of pixels within the cell's mask. This pixel count is then converted into a physical area (e.g., in square micrometers, μm^2) using the known scale of the image.
        2. **Elongation:** A robust and quantitative measure of cell shape is eccentricity. This is calculated by fitting an ellipse to the cell's mask. The eccentricity of this ellipse, a value ranging from 0 for a perfect circle to 1 for a line segment, provides a continuous measure of elongation. This allows for the quantitative differentiation between the "round" and "oval" cell phenotypes mentioned in the research context.
        3. **Clumping:** Groups of adherent cells can be identified by analyzing the spatial relationships between the instance masks. A simple approach is to compute a proximity graph where nodes are cells and an edge exists if their bounding boxes are within a certain distance or if their masks overlap. Connected components in this graph with more than one node represent cell clumps.

4.2 Biofilm Characterization: Quantifying Hyphal Morphology

The objective of this task is to quantify the structural characteristics of the hyphal networks within the biofilm, including properties like filament length, branching complexity, and tortuosity (or "twistiness").

- **Methodology: Skeletonization and Graph Analysis.** This task requires analyzing the overall shape and topology of the biofilm, rather than individual cells. The binary biofilm mask produced by the U-Net model in Phase II serves as the ideal input for this analysis.
    - **Pipeline:**
        1. **Skeletonization:** A morphological skeletonization algorithm (e.g., the Zhang-Suen algorithm) is applied to the binary biofilm mask. This iterative thinning process erodes the boundaries of the foreground region while preserving its topology, reducing the thick, multi-pixel-wide hyphal structures down to one-pixel-wide centerlines. The result is a "skeleton" that acts as a graph-like representation of the hyphal network.
        2. **Graph Extraction:** The resulting skeleton image is then converted into a formal graph data structure (e.g., using a library like skan or NetworkX). In this graph, pixels in the skeleton become nodes, and connections between adjacent pixels become edges.
        3. **Feature Calculation:** Once the network is represented as a graph, standard graph traversal algorithms can be used to compute biologically relevant morphological features:
            - **Hyphal Length:** The length of each individual hyphal filament can be calculated by measuring the path length of each branch (an edge or series of edges between two junction points or a junction point and an endpoint) in the graph.
            - **Branching:** Branch points in the hyphal network correspond to nodes in the graph with a degree of 3 or more (i.e., a pixel connected to three or more other pixels). The total number of such nodes provides a measure of branching complexity.
            - **Twistiness (Tortuosity):** For each hyphal segment (a path between two endpoints/junctions), its tortuosity can be calculated as the ratio of its actual path length (sum of edge lengths) to the straight-line Euclidean distance between its start and end nodes. A value close to 1 indicates a straight filament, while higher values indicate a more convoluted or twisted path.

4.3 Cell Tracking: Analyzing Post-Dispersal Dynamics

The objective of this final advanced task is to track individual dispersed cells across consecutive frames. This enables the analysis of their behavior over time, such as quantifying their post-dispersal growth rates or motility patterns, which is a key goal of the research proposal.

- **Methodology: DeepSORT (Deep Simple Online and Realtime Tracking).** For multi-object tracking in potentially crowded scenes with occlusions, a sophisticated algorithm is required. DeepSORT is an excellent choice as it extends the popular SORT tracker by integrating a deep learning-based appearance model, making it significantly more robust. While SORT relies solely on motion information (predicting where a box will be) and overlap (IoU), DeepSORT adds a crucial third component: what the object looks like. This dramatically reduces identity switches, where the tracker mistakenly swaps the IDs of two objects that pass close to each other.

- **Implementation Plan:**
    1. **Detection Input:** The tracker will operate on a frame-by-frame basis, taking the set of bounding boxes (or masks) from the instance segmentation model (Task 4.1) as the input detections for each frame.
    2. **Motion Model:** For each tracked object, a Kalman filter is used to model its motion. The Kalman filter maintains a state for each object (typically including position, velocity, and box dimensions) and predicts its state in the next frame. This prediction provides a motion-based estimate of where to look for the object.
    3. **Appearance Model:** This is the core innovation of DeepSORT. A small Convolutional Neural Network (CNN) is trained specifically to extract a compact feature vector, or "embedding," from an image patch of a detected cell. This embedding serves as a unique appearance signature. The CNN is trained on a dataset of cropped images of individual cells, learning to produce similar embeddings for images of the same cell and different embeddings for images of different cells.
    4. **Data Association:** In each new frame, the tracker must associate the new detections with the existing tracks. This is formulated as an assignment problem, solved using the Hungarian algorithm. The cost matrix for this assignment is a weighted combination of two metrics:
        - **Motion Cost:** The Mahalanobis distance between a new detection and a track's predicted position from the Kalman filter. This measures how plausible the match is from a motion perspective.
        - **Appearance Cost:** The cosine distance between the appearance embedding of a new detection and the historical embeddings stored for an existing track. This measures how visually similar the detection is to the tracked object.
    5. **Output Generation:** The final output of the DeepSORT module is a set of trajectories. Each trajectory consists of a unique track ID and a list of bounding box coordinates for that specific cell in every frame in which it was successfully tracked. This structured output directly enables the analysis of post-dispersal growth. For example, one can plot the size (from Task 4.1) of the cell associated with track_ID_123 over time to generate an individual cell growth curve.

Section 5: Phase IV - Pipeline Integration, Final Evaluation, and Reporting (Weeks 14-15)

The final phase of the project is dedicated to consolidating the developed modules into a cohesive, automated system, rigorously evaluating its performance, and meticulously preparing the final submission materials. This phase transitions the project from a collection of experimental components into a polished, production-ready solution that effectively communicates its technical innovations and scientific contributions.

5.1 System Integration: Creating an Automated End-to-End Workflow

The primary objective of this stage is to assemble all the individual modules developed in Phases II and III into a single, executable pipeline. This pipeline should take a raw time-lapse video as input and automatically generate all the required quantitative data, analyses, and visualizations as output, without manual intervention.

- **Workflow Orchestration:** To achieve this, a pipeline orchestration tool such as Vertex AI Pipelines or AWS Step Functions will be used. These tools allow for the definition of a workflow as a Directed Acyclic Graph (DAG), where each node in the graph represents a processing step (e.g., a script or a containerized application) and the edges define the flow of data and dependencies between them. The integrated pipeline will be structured as follows:
    1. **Input:** Raw time-lapse image sequence.
    2. **Step 1 (Pre-processing):** Apply the trained Super-Resolution model to each frame.
    3. **Step 2 (Parallel Processing):**
        - **Branch A:** Feed the super-resolved images to the trained U-Net model to generate biofilm segmentation masks.
        - **Branch B:** Feed the super-resolved images to the trained instance segmentation model to generate individual cell masks and bounding boxes.
    4. **Step 3 (Parallel Analysis):**
        - **Branch A:** Process the biofilm masks through the skeletonization and graph analysis module to extract hyphal characteristics.
        - **Branch B:** Process the individual cell masks through the morphological feature extraction module to calculate cell size and elongation.
        - **Branch C:** Feed the sequence of bounding boxes from Step 2B into the DeepSORT module to generate cell trajectories.
    5. **Step 4 (Final Synthesis):**
        - Aggregate the outputs to generate the final quantitative results: biofilm growth curve, dispersed cell counts, dispersal initiation time point, distributions of cell/biofilm features, and cell tracking data.
        - Generate all required plots and visualizations.

This automated pipeline is not just a convenience; it is a key deliverable that demonstrates a mature, production-ready system and ensures the final results are fully reproducible.

5.2 Final Evaluation and Hyperparameter Tuning at Scale

With the integrated pipeline in place, the final step before generating the submission results is to optimize the performance of the entire system. Manually tuning the numerous hyperparameters of multiple deep learning models is inefficient and unlikely to find the optimal configuration.

- **Methodology:** Cloud-based automated hyperparameter tuning services, such as Vertex AI Vizier or Amazon SageMaker Automatic Model Tuning, will be leveraged. These services use sophisticated search algorithms (e.g., Bayesian optimization) to intelligently and efficiently explore the vast hyperparameter space. Separate tuning jobs will be configured for the most critical models in the pipeline: the U-Net for biofilm segmentation, the instance segmentation model for cell detection and characterization, and the appearance CNN within the DeepSORT tracker. This systematic, large-scale tuning process will ensure that the final models used for the submission are operating at their peak performance.

5.3 A Roadmap to Final Deliverables

The final task is to package the project's outcomes into a clear, compelling, and comprehensive submission that maximizes its impact and score.

- **Final Report Structure:** The written report will be the primary document explaining the

project's methodology and results. It will be structured to tell a coherent story:
    - **Introduction:** Begin by framing the problem in its rich biological context, drawing heavily from the project description and the provided research paper to establish the scientific significance of analyzing *C. albicans* dispersal.
    - **Methods:** Detail the four-phase strategic approach. For each analytical task (e.g., biofilm segmentation, cell tracking), clearly describe the chosen methodology and provide a concise but rigorous justification for that choice, potentially referencing the "Algorithmic Strategy and Justification" table. A key focus will be explaining the "Enhance-then-Analyze" strategy as the central solution to the low-magnification challenge.
    - **Results:** Systematically present the final, optimized outputs for each evaluation criterion, starting with the minimal model and progressing to the "icing on the cake." This section must be rich with high-quality visualizations: the final biofilm growth curve, plots of dispersed cell counts over time with the detected change point clearly marked, histograms showing the distribution of cell sizes and elongations, and example images overlaid with segmentation masks and tracking IDs.
    - **Discussion:** Go beyond simply presenting the results by interpreting them in a biological context. Discuss what the shape of the growth curve implies about biofilm maturation. Analyze the distribution of dispersed cell phenotypes and relate it back to the concept of subpopulations mentioned in the research. Discuss any observed patterns in post-dispersal cell behavior revealed by the tracking data.
    - **Conclusion & Future Work:** Succinctly summarize the project's key achievements and technical innovations. Suggest potential avenues for future work, such as applying the pipeline to different experimental conditions or extending the characterization to include more complex features.
- **Final Presentation:** The presentation will be a visual and dynamic summary of the project. It should follow a similar narrative arc to the report but focus on high-impact visuals. Video clips showing the final tracking output, with cells correctly maintaining their unique IDs as they move and interact, will be a particularly powerful way to demonstrate the system's capabilities.
- **Code Submission:** The final codebase must reflect the professionalism established by the MLOps foundation. The code should be clean, well-commented, and organized logically into modules. A comprehensive README.md file is essential. It must provide clear, step-by-step instructions on how to set up the environment and run the entire integrated pipeline to reproduce the results presented in the report. This final step closes the loop on reproducibility and showcases a complete, well-engineered solution.

Works cited

1. Understanding Clearly the Magnification of Microscopy - Leica Microsystems, https://www.leica-microsystems.com/science-lab/industrial/understanding-clearly-the-magnification-of-microscopy/ 2. Microscopy 101: Understanding Magnification, Resolution, and Lenses - Motic Microscopes, https://moticmicroscopes.com/blogs/articles/microscopy-101-understanding-magnification-resolution-and-lenses 3. Optical and digital microscopic imaging techniques and applications in pathology - PMC, https://pmc.ncbi.nlm.nih.gov/articles/mid/NIHMS363450/ 4. Microscopy Image Analysis « - College of Engineering - Purdue University, https://engineering.purdue.edu/~micros/ 5. A Complete Guide to Image Super-Resolution in Deep Learning and AI | DigitalOcean,

https://www.digitalocean.com/community/tutorials/image-super-resolution 6. Lightweight Super-Resolution Techniques in Medical Imaging: Bridging Quality and Computational Efficiency - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC11673497/ 7. Deep learning for enhancement of low-resolution and noisy scanning probe microscopy images - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC12278107/ 8. impact of low resolution on the image recognition with deep neural networks: an experimental study - AGH, https://home.agh.edu.pl/~cyganek/impact-low-resolution.pdf 9. 4 No Cost Ways To Improve Your Microscopy Image Quality - ExpertCytometry, https://expertcytometry.com/4-no-cost-ways-to-improve-your-microscopy-image-quality/ 10. Quantifying microscopy images: top 10 tips for image acquisition - Carpenter-Singh Lab, https://carpenter-singh-lab.broadinstitute.org/blog/quantifying-microscopy-images-top-10-tips-for-image-acquisition 11. Train Machine Learning Models – Amazon SageMaker Model Training - AWS, https://aws.amazon.com/sagemaker/ai/train/ 12. Revolutionizing AI and Data Management with IBM WatsonX – Blog, https://blog.miraclesoft.com/revolutionizing-ai-and-data-management-with-ibm-watsonx/ 13. Introduction to Vertex AI | Google Cloud, https://cloud.google.com/vertex-ai/docs/start/introduction-unified-platform 14. Build, Deploy, and Manage ML Models with Google Vertex AI - Analytics Vidhya, https://www.analyticsvidhya.com/blog/2024/02/build-deploy-and-manage-ml-models-with-google-vertex-ai/ 15. Vertex AI Chronicles: Exploring End-to-End ML Solutions - NashTech Blog, https://blog.nashtechglobal.com/vertex-ai-chronicles-exploring-end-to-end-ml-solutions/ 16. SageMaker Studio: ML Development Overview - AWS, https://aws.amazon.com/awstv/watch/faa2c877499/ 17. Transforming Machine Learning with Google's Vertex AI Solutions, https://blog.miraclesoft.com/transforming-machine-learning-with-googles-vertex-ai/ 18. Machine Learning Operations Tools - Amazon SageMaker for MLOps, https://aws.amazon.com/sagemaker/ai/mlops/ 19. IBM watsonx, https://www.ibm.com/products/watsonx 20. Building Scalable AI Pipelines with Vertex AI on Google Cloud - CLOUDSUFI, https://www.cloudsufi.com/building-scalable-ai-pipelines-with-vertex-ai-on-google-cloud/ 21. U-Net - Wikipedia, https://en.wikipedia.org/wiki/U-Net 22. [1505.04597] U-Net: Convolutional Networks for Biomedical Image Segmentation - arXiv, https://arxiv.org/abs/1505.04597 23. U-Net Architecture For Image Segmentation | DigitalOcean, https://www.digitalocean.com/community/tutorials/unet-architecture-image-segmentation 24. U-Net: A Versatile Deep Learning Architecture for Image Segmentation - Medium, https://medium.com/@alexquesada22/u-net-a-versatile-deep-learning-architecture-for-image-segmentation-2a85b52d71f6 25. U-Net: Convolutional Networks for Biomedical Image Segmentation, https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/ 26. U-Net-Based Segmentation of Microscopic Images of Colorants and Simplification of Labeling in the Learning Process - MDPI, https://www.mdpi.com/2313-433X/8/7/177 27. (PDF) Automatic Cell Counting With YOLOv5: A Fluorescence Microscopy Approach, https://www.researchgate.net/publication/373744977_Automatic_Cell_Counting_With_YOLOv5_A_Fluorescence_Microscopy_Approach 28. YOLOv5-FPN: A Robust Framework for Multi-Sized Cell Counting in Fluorescence Images, https://pmc.ncbi.nlm.nih.gov/articles/PMC10341068/ 29. Automatic Cell Counting With YOLOv5: A Fluorescence Microscopy Approach - Documat, https://documat.unirioja.es/descarga/articulo/9441896.pdf 30. How Change Point Detection works—ArcGIS Pro | Documentation,

https://pro.arcgis.com/en/pro-app/latest/tool-reference/space-time-pattern-mining/how-change-point-detection-works.htm 31. Multivariate Time Series Change-Point Detection with a Novel Pearson-like Scaled Bregman Divergence - MDPI, https://www.mdpi.com/2571-905X/7/2/28 32. Real-Time Change Point Detection with application to Smart Home Time Series Data - School of Electrical Engineering & Computer Science, https://eecs.wsu.edu/~cook/pubs/tkde18.pdf 33. Introduction to optimal changepoint detection algorithms - R Project, https://www.r-project.org/conferences/useR-2017//uploads/TobyHocking.html 34. Single-cell bioimage analysis for feature extraction. a-d Performance... - ResearchGate, https://www.researchgate.net/figure/Single-cell-bioimage-analysis-for-feature-extraction-a-d-Performance-demonstration-of_fig5_331982146 35. Cell Imaging Feature Extraction and Morphology Clustering for Spatial Omics | NVIDIA Technical Blog, https://developer.nvidia.com/blog/cell-imaging-feature-extraction-and-morphology-clustering-for-spatial-omics/ 36. Features — polarityjam documentation - Read the Docs, https://polarityjam.readthedocs.io/en/latest/Features.html 37. Multiple Object Tracking Based on YOLOv5 and Optimized DeepSORT Algorithm, https://www.researchgate.net/publication/372904860_Multiple_Object_Tracking_Based_on_YOLOv5_and_Optimized_DeepSORT_Algorithm 38. Deep SORT: Realtime Object Tracking Guide - Ikomia, https://www.ikomia.ai/blog/deep-sort-object-tracking-guide 39. Improving Cell Detection and Tracking in Microscopy Images Using YOLO and an Enhanced DeepSORT Algorithm - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC12297877/ 40. Understanding Multiple Object Tracking using DeepSORT - LearnOpenCV, https://learnopencv.com/understanding-multiple-object-tracking-using-deepsort/ 41. Sort and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation with New Data Association Metrics - MDPI, https://www.mdpi.com/2076-3417/12/3/1319 42. Multi-Object Tracking with DeepSORT - MATLAB & Simulink - MathWorks, https://www.mathworks.com/help/fusion/ug/multi-object-tracking-with-deepsort.html