

SOFTWARE TOOL FOR LENSELESS NANO FIBER OPTIC ILLUMINATION MICROSCOPE

Sachin G Lokapure¹, Abhinandan R Patil^{1*}, Chandraprabhu Jangme¹

¹Centre for Interdisciplinary Research, D. Y. Patil Education Society, (Deemed to be University) Kolhapur 416006, Maharashtra, India

Corresponding author: Dr. Abhinandan Patil | D. Y. Patil Education Society, (Deemed to be University) Kolhapur 416006, Maharashtra, India | Email: abhisirdyp@gmail.com

ABSTRACT

The "Software Tool for Lenseless Microscopy" is developed to facilitate efficient acquisition, processing, and analysis of biological samples using a compact, lens-free imaging system. This software integrates multiple functional modules that collectively enhance the usability and precision of digital microscopy. Designed for researchers and laboratory professionals, the system bridges the gap between hardware components—like CMOS sensors—and image processing frameworks through a seamless software interface.

The application begins with secure login access for both users and administrators. Admins are granted functionalities such as user management and system activity monitoring through logs. For regular users, the software enables direct image capture via connection to a CMOS sensor, acquisition of raw images, and subsequent image enhancement using OpenCV. The processed images are displayed in real-time and can be stored locally for further analysis or documentation.

Additional features include automatic logging of user activities, viewing and exporting of analytical reports, and the ability to print high-resolution output. By combining image acquisition, processing, storage, and reporting into a single platform, the tool ensures accuracy, efficiency, and ease of use for laboratory applications.

This software aims to significantly streamline laboratory workflows, reduce dependency on bulky traditional microscopes, and provide a cost-effective, scalable solution for biological imaging. Its modular architecture, real-time processing capabilities, and intuitive user interface make it a vital tool in modern digital pathology and biomedical research environments.

Keywords: Python, Image Processing, CMOS Sensors, USB Integration.

How to cite this article: Lokapure SG, Patil AR, Jangme C. Software Tool for Lensless Nano Fiber Optic Illumination Microscope. *Int J Drug Deliv Technol.* 2026;16(54s): 1096-1120. DOI: 10.25258/ijddt.16.54s.95

Source of support: Nil.

Conflict of interest: None.

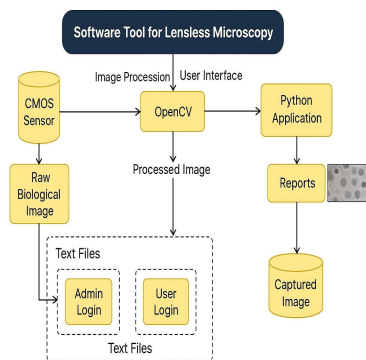
Introduction

In today's rapidly advancing scientific era, there is an increasing demand for innovative imaging technologies, especially in the domains of biological and medical research. In fields such as molecular biology, pathology, and clinical diagnostics, the ability to visualize and analyze biological samples with precision is critical for understanding cellular structures, disease progression, and treatment efficacy. Traditional optical microscopy, while highly effective, often involves expensive lenses, complex equipment, and bulky setups that may not be suitable for all laboratory environments—especially in resource-constrained settings. To address these limitations, our project introduces a cost-effective and portable solution: a software tool for lenseless microscopy. This tool is designed to enable high-resolution biological

imaging using a CMOS sensor and advanced computational methods, eliminating the need for conventional optical lenses. [1-6] The core technology behind lenseless microscopy involves capturing light diffraction patterns from biological samples using a CMOS sensor and reconstructing detailed images through sophisticated image processing algorithms. Our project employs phase retrieval, de-convolution, and denoising techniques implemented using Python libraries such as OpenCV and NumPy. These algorithms interpret the raw sensor data and reconstruct accurate images, offering researchers valuable insights at a lower cost and greater accessibility. The simplicity of the hardware design—relying primarily on the CMOS sensor and Raspberry Pi 4 B+—makes this solution ideal for deployment in educational, field, or under-resourced laboratory environments. [7] The software tool features a user-

RESEARCH PAPER

friendly graphical interface built using PyQt5, allowing users to easily interact with the system. The interface supports functions such as real-time image capture, visualization, image enhancement, and PDF report generation using ReportLab. The inclusion of user and admin login functionality ensures controlled access, while an activity log records system usage for transparency and traceability. The application runs on various platforms, making it accessible on standard desktop systems. [8] A secure and structured file-based storage system is integrated to manage captured images and their corresponding analysis results. This facilitates easy retrieval, comparison, and validation of previous data, enhancing the research workflow. Researchers can upload new samples, execute image processing algorithms, and generate detailed reports without requiring extensive training or technical knowledge. This seamless integration of hardware and software promotes faster experimentation, real-time feedback, and more efficient decision-making. To ensure real-time performance, the system is optimized to process and



display captured images immediately, making it suitable for dynamic scenarios such as live-cell imaging and time-sensitive diagnostics. The tool is also designed to be modular, allowing for future extensions—such as automated sample detection or integration with external lab equipment. In summary, the Software Tool for Lenseless Microscopy provides a compact, low-cost, and powerful alternative to traditional microscopy systems. By leveraging computational optics,

sensor-based imaging, and modern software development techniques, this project contributes significantly to democratizing access to advanced biological imaging for educational institutions, research labs, and clinics worldwide. [9-10]

1. Material and method

1.1 Methodology

The development of the Software Tool for Lenseless Microscopy followed a modular and iterative approach to ensure accuracy, usability, and performance. The methodology is broken down into the following phases:

- **Requirement Analysis:**

Before initiating the project, a thorough requirement analysis was conducted. The goal was to build a cost-effective and portable microscopy solution that could capture, process, and visualize microscopic images using a CMOS sensor connected to a system. Key functional requirements included:

1. Real-time image capture using CMOS sensor hardware.
2. Preprocessing and enhancement using OpenCV.
3. An interactive desktop-based GUI developed in PyQt5.
4. Secure user and admin login systems.
5. Activity logging and image export functionalities.
6. Report generation in PDF format using ReportLab.

Figure 1: Architecture of Software Tool for Lenseless Microscopy.

- **System Design and Architecture:**

The system was developed using a modular, scalable architecture tailored for image acquisition,

RESEARCH PAPER

processing, and reporting. The design emphasizes separation of concerns across its major components to ensure maintainability and performance. [11]

The architecture follows a three-tier structure:

Presentation Layer: This layer is implemented using PyQt5 and provides an intuitive graphical interface for both regular users and administrators. The interface includes panels for live image capture, real-time processing options, report generation, and user management. Designed with accessibility in mind, it allows users with minimal technical background to navigate features such as image filters, zoom tools, and report saving. Status updates, logs, and feedback are also displayed in real time to guide user interaction.

Application Layer: Acting as the core logic handler, this layer bridges the GUI with underlying image processing and data handling modules. It handles user authentication, controls the CMOS sensor interface for capturing images, applies processing operations using OpenCV, and generates structured reports using ReportLab. The logic is designed to handle sequential operations such as initializing camera streams, processing frames, and storing session data while maintaining responsiveness. This layer also manages error handling and file management, ensuring a robust user experience.

Data Storage Layer: The system uses a structured file-based storage mechanism to organize captured images, logs, and PDF reports. Each user session is timestamped and saved in uniquely identified folders, maintaining traceability of samples and operations. Though a lightweight design is adopted, the system

architecture is scalable enough to support database integration in future upgrades, enabling features like indexed search and centralized records. To ensure future adaptability and maintain loose coupling between modules, the system logic is encapsulated in callable classes and functions, allowing for potential API integration. While current communication is internal and event-driven, the design is prepared for REST API inclusion, enabling future enhancements like remote monitoring, mobile app integration, or cloud-based backup. The architecture also focuses on resource efficiency and portability. It is optimized to run on low-powered systems such as Raspberry Pi-based setups, ensuring feasibility in educational and remote lab environments. The system captures high-resolution images while keeping memory usage minimal, allowing smooth performance even without high-end hardware. Security is embedded through file access restrictions and role-based login controls, ensuring data confidentiality and preventing unauthorized changes or access to sensitive images and reports. While no external network communication is currently involved, the system design adheres to principles of local data privacy and user integrity. [12]

- **Module Development:** Each module in the Software Tool for Lenseless Microscopy is purposefully designed to streamline the process of capturing, processing, analyzing, and reporting microscopic images. The modular architecture allows independent development, testing, and integration of features, making the system flexible and maintainable.

Image Acquisition Module

RESEARCH PAPER

serves as the system's entry point, interfacing directly with the CMOS image sensor to capture raw data. This module handles camera initialization, configuration of capture parameters such as exposure and resolution, and real-time display of live imaging. It is designed to support plug-and-play operation, allowing users to initiate sample capture with minimal setup or technical knowledge.

Image Processing Module is responsible for enhancing and analyzing captured images using OpenCV. It offers a suite of tools including noise reduction, contrast adjustment, grayscale conversion, and edge detection. These operations are essential for improving the visibility of microscopic features in biological samples. The module supports both automatic and manual modes, enabling users to apply custom filters or rely on pre-configured processing pipelines for consistent results.

User Management Module facilitates secure access to the system. It includes role-based authentication for administrators and general users. Administrators can access system logs, manage stored data, and oversee system settings, while users are limited to image capture and report generation. This division enhances data security and operational control, particularly in shared lab environments.

Session Logging and History Module maintains a detailed record of all user activities and image captures. Each session is timestamped and stored in structured folders for easy retrieval and auditing. This module ensures reproducibility of experiments and provides traceability in academic or clinical research contexts. It

also contributes to system reliability by logging errors and operational events for diagnostics.

PDF Report Generation Module automates the creation of structured reports summarizing each microscopy session. Reports include processed images, metadata such as user ID, date, and sample description, along with applied processing techniques. Using the ReportLab library, the system generates professional-quality documents suitable for academic documentation, project submissions, or lab records.

File Management Module organizes and safeguards image and report data. Captured images and their processed versions are stored in separate directories for clarity. The module also prevents unintentional overwriting and supports basic file operations like renaming, deletion, and backup preparation. Future upgrades can link this module with cloud storage or centralized databases for enhanced accessibility.

Together, these modules work cohesively to create an end-to-end Lenseless microscopy solution that is low-cost, portable, and effective. Each component is crafted to support the broader goals of accessibility, accuracy, and efficiency in biological imaging, making the system suitable for educational, research, and resource-limited healthcare settings. [13]

- **Database Integration:** To manage user authentication, session logs, and metadata associated with captured and processed images, the Software Tool for Lenseless Microscopy employs a lightweight yet efficient file-based data storage approach. Given the standalone and offline nature of the system, especially in educational or

RESEARCH PAPER

research settings where internet access might be limited, this method ensures simplicity and reliability without the overhead of a full-scale database server.

The storage system organizes data into structured directories that correspond to user sessions and image capture activities. Each session folder contains the original and processed image files, along with associated metadata stored in structured text or JSON files. These metadata records include details such as user ID, capture date and time, sample description, and image processing parameters applied. This structured format enables easy retrieval, indexing, and sorting of session data during report generation or when reviewing past activities.

For authentication and access control, a secure file-based mechanism maintains user credentials and access roles. Each user has a defined role, such as administrator or general user, which determines the level of access to different functionalities within the software. Role-based restrictions are enforced during login, ensuring that sensitive features like data deletion, session logs, and system configurations are accessible only to authorized personnel.

Although not built on a traditional relational database, the system is designed to be scalable by maintaining consistent directory structures and naming conventions. This logical organization allows users to handle increasing volumes of data without performance degradation. Furthermore, each user session is isolated in its own folder, making data backup and portability straightforward. For larger deployments or future upgrades, the system design can be easily adapted to support SQLite or MySQL for

enhanced data handling.

Data security is a core consideration in the file management system. Access to data directories is restricted based on user roles, and file permissions are set to prevent unauthorized modification or deletion of critical records. In addition, sensitive user credentials are stored using hashing techniques to protect against unauthorized access. Transaction integrity is ensured through the use of intermediate storage buffers. When image data is being processed or a PDF report is being generated, temporary files are used until the operation is confirmed as successful. Only after validation are final files saved to the permanent session folder. This prevents corruption of data due to unexpected errors or interruptions during processing.

To maintain audit trails, the system logs user activity, including login timestamps, image captures, processing actions, and report generations. These logs are stored in a centralized log file or directory, helping administrators monitor usage patterns and troubleshoot issues. Such tracking also provides a level of accountability important in research environments, especially when the software is used in collaborative settings.

While the current file-based approach offers simplicity and efficiency for desktop use, the system architecture is designed with extensibility in mind. In future iterations, it can be seamlessly integrated with cloud-based storage solutions or upgraded to use a relational database to support multi-user environments and remote data access. This flexibility ensures that the Software Tool for Lenseless Microscopy can

RESEARCH PAPER

evolve with user needs while maintaining its core values of accessibility, security, and performance. [14]

- **User Interface Design:** The user interface (UI) of the Software Tool for Lenseless Microscopy is developed using PyQt5, a Python-based GUI framework that enables the creation of cross-platform desktop applications with rich, native-style interfaces. The interface is designed to be clean, intuitive, and user-friendly, ensuring that users from research, academic, or clinical backgrounds can navigate the software with ease, regardless of their technical expertise.

The main dashboard provides easy access to all core functionalities, including image capture, processing, user login management, activity logs, and report generation. Each module is organized into dedicated tabs or windows, reducing clutter and ensuring that users can focus on a single task at a time. The layout is consistent throughout the application, with clearly labeled buttons, tooltips, and visual feedback mechanisms that guide the user through each step of the image acquisition and analysis workflow.

The design prioritizes clarity and responsiveness. Icons and graphical elements are chosen for simplicity and functionality, ensuring a minimal learning curve. Features such as real-time image display, zoom options, and enhancement sliders are integrated into the interface to offer a hands-on, interactive experience while viewing and analyzing captured images. This helps users immediately see the results of processing techniques such as contrast adjustment or edge detection.

The UI also incorporates login screens for administrators and general users, with tai-

lored access to specific features based on their roles. Administrators can access and manage session logs, configure system settings, and oversee user activity, while general users are provided with tools for capturing images and generating reports. This role-based customization ensures that each user has a focused and relevant interface without unnecessary distractions.

Although primarily a desktop application, the interface is designed with scalability in mind. It can be adapted for touch-enabled devices or ported to a web interface in future versions, offering greater flexibility in deployment. Visual consistency is maintained across different operating systems, providing a familiar experience on Windows, Linux, or macOS environments.

- **Real-Time Data Processing:** One of the key capabilities of the Software Tool for Lenseless Microscopy is its ability to perform real-time data processing to facilitate immediate visualization and analysis of biological samples. The system captures raw image data from a CMOS sensor and processes it on the fly using Python libraries such as OpenCV. This allows users to observe processed outputs, such as enhanced contrast or noise reduction, in near real-time as the image is being captured.

When a user initiates the image acquisition process, the application immediately displays the live feed from the sensor, applying processing algorithms like sharpening, thresholding, and filtering in real time. This immediate feedback helps users determine the quality of the captured image without the need for additional steps, significantly improving the efficiency of

RESEARCH PAPER

microscopic examination in laboratory or field settings.

The system also integrates real-time status indicators for hardware connectivity and image processing stages, ensuring that users are continuously informed of system operations. If any error occurs during image acquisition or processing, appropriate alerts are displayed instantly, allowing users to take corrective action without interrupting the workflow.

Moreover, activity logs are updated in real time, maintaining a continuous record of user operations, captured images, and analysis results. This ensures that all user interactions and system responses are tracked instantly for auditing and troubleshooting purposes.

By offering real-time image capture and processing, the tool significantly enhances productivity and usability, particularly in scenarios where immediate sample evaluation is necessary. This feature ensures that users can make timely decisions based on visual data, supporting faster diagnostics, research, and educational outcomes. [15]

- **Testing and Quality Assurance:** Testing plays a critical role in ensuring that the Software Tool for Lenseless Microscopy is stable, functional, and ready for deployment in research and educational environments. The project undergoes multiple levels of testing to validate its performance, reliability, and usability across all features and modules.

Unit Testing is performed on individual components such as image capture, real-time processing, user login, and PDF report generation. Each function is tested independently to ensure that it operates correctly under

different scenarios. This includes verifying that the CMOS sensor captures images accurately and that image processing filters like contrast adjustment, thresholding, and edge detection work as intended.

Integration Testing is used to confirm that different modules, including the GUI, image acquisition, processing pipeline, and file storage, work cohesively. It ensures that data captured by the sensor is correctly processed, displayed, and saved, and that reports are generated from the correct input sources without any data loss or misalignment.

Performance Testing evaluates the system's ability to handle large image files and continuous data streams from the CMOS sensor. It verifies that the application remains responsive during real-time processing and that the graphical user interface remains stable even when performing complex operations or generating high-resolution reports.

Security Testing is conducted to ensure that sensitive data, such as user credentials and activity logs, is securely stored and protected against unauthorized access. This includes validating password protection, secure file storage practices, and limiting access to administrative functionalities based on user roles.

User Acceptance Testing (UAT) is carried out with researchers and students who represent the target user base. Feedback is collected on ease of use, functionality, and clarity of the interface. Suggestions from UAT help refine the interface layout, optimize tool responsiveness, and ensure that the application aligns with user expectations for real-world microscopy

RESEARCH PAPER

applications.

Through rigorous and iterative testing, the system is refined and validated to ensure it performs reliably in practical settings, supporting accurate imaging, user-friendly interaction, and data integrity across its operations.

- **Deployment and Feedback:**

Once the Software Tool for Lenseless Microscopy successfully passed all testing phases, it was deployed in a controlled environment within a research or educational laboratory for real-world evaluation. This initial deployment phase allowed the development team to observe how the application performs when used by students and researchers for live sample imaging and analysis. It provided an opportunity to uncover practical issues that may not have surfaced during internal testing. Feedback was collected through direct user interactions, structured questionnaires, and usage analytics to identify usability bottlenecks and technical gaps.

The pilot deployment helped evaluate the efficiency of core features like image capture from the CMOS sensor, real-time processing, and PDF report generation. Observations and feedback during this stage were crucial in highlighting areas where users faced delays or interface difficulties. The development team actively monitored the system's stability and responsiveness during extended use to ensure that processing modules and user workflows operated as intended.

To improve post-deployment monitoring, logs and diagnostic tools were integrated to track processing times, interface actions, and error occurrences. These logs helped in identifying areas where optimization was needed, such as minimizing

latency in image rendering or improving memory management during image processing.

Researchers and students participating in the pilot were encouraged to share detailed feedback on the user experience, including ease of navigation, clarity of output, and utility of the PDF reports. Based on this feedback, updates were prioritized to improve functionality, streamline navigation, and enhance report customization.

Following the pilot deployment, refinements were made to address identified issues, which included optimizing the layout of the interface, improving feedback messages, and fine-tuning the accuracy of image filters. After implementing these improvements, the system was prepared for broader adoption in similar laboratory settings. Continuous feedback is encouraged to maintain an iterative development approach, ensuring that the system evolves based on real-world needs and provides lasting value to its users.

- **Ongoing Support and Maintenance:**

To ensure the effective adoption of the Software Tool for Lenseless Microscopy, comprehensive training and support mechanisms are put in place. These include user manuals, step-by-step documentation, and instructional videos aimed at helping students, researchers, and laboratory technicians understand the system's functionalities. The training resources are designed to accommodate users with varying levels of technical expertise, ensuring ease of use across the board.

Role-based guidance is provided to ensure that different user groups—such as system administrators, educators, and end-users—receive relevant training for

RESEARCH PAPER

their specific roles. For example, administrators are trained on user management, data handling, and system configuration, while researchers and students focus on image capture, real-time processing, and report generation features.

Interactive training sessions are also conducted in a simulated environment to familiarize users with the interface and workflow before full deployment. These sessions reduce the learning curve and instill confidence in using the system effectively. Furthermore, ongoing support is available through documentation and developer contact to assist users in resolving technical issues or clarifying usage queries during system operation. To ensure the scalability and adaptability of the Software Tool for Lenseless Microscopy, the system is built with a modular architecture. This allows for independent development and maintenance of core components such as image acquisition, real-time processing, PDF report generation, and user authentication. The modular approach makes it easy to integrate future enhancements, such as AI-based classification or cloud-based data storage, without affecting the stability of existing functionalities. The tool's real-time image processing capability ensures that captured samples are processed and displayed immediately, enhancing productivity in laboratory environments. Users can visualize results on-the-fly, analyze sample quality, and recapture images if necessary. This real-time feedback loop significantly improves the efficiency of sample analysis workflows, especially in educational or research settings where timely results

are crucial. The user interface is developed using PyQt5, offering a responsive and consistent experience across different screen sizes. The UI is designed with a focus on usability and accessibility, allowing students and researchers to navigate the application with minimal effort. Features such as progress indicators, file selection dialogs, and image previews are integrated to enhance user interaction and reduce operational complexity. Data security and integrity are carefully considered in the design of the application. While the tool operates using a file-based data storage system instead of a centralized database, safeguards are implemented to ensure sensitive data such as user activity logs and sample records are securely stored. Role-based access is enforced, with login credentials limiting access to administrative features and personal reports, ensuring user accountability and data privacy. The system is subjected to comprehensive testing procedures including unit testing of individual modules, integration testing for seamless interaction between components, and user acceptance testing (UAT) with students and researchers. This ensures that the application is robust, functional, and well-suited for real-world usage. Feedback collected from users post-deployment helps identify pain points and areas for enhancement, guiding future development efforts. Overall, the Software Tool for Lenseless Microscopy is designed to be a reliable, secure, and user-friendly desktop solution for capturing, processing, and analyzing biological samples using a CMOS sensor. Its real-time capabilities, modular design, and role-based access control make it ideal for laboratory and educational use. With

RESEARCH PAPER

continuous improvements driven by user feedback and an emphasis on performance and usability, the system is poised to support low-cost microscopy needs in both research and academic domains. [17-20]

1.2 UML Diagrams

1.2.1 Use Case Diagram

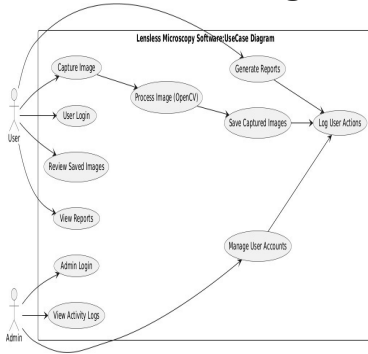


Figure 2: Usecase Diagram of Software Tool for Lenseless Microscopy.

The use case diagram for the Lenseless Microscopy Software illustrates the core functionalities of the system and how different users interact with it. The primary actors involved are the User and the Admin. Users can log in, capture images using a Lenseless microscope setup, and process those images using OpenCV tools. Once processed, images can be saved and later reviewed. Users can also generate detailed reports based on the captured data and view previously created reports. Each of these tasks is designed to streamline the user's workflow while ensuring that image data is properly handled and stored.

On the administrative side, the Admin actor has access to additional functionalities. After logging in, the admin can manage user accounts, monitor the activity logs of all users, and ensure system integrity. Admins play a crucial role in overseeing the system's operations and maintaining security. The use case diagram also includes system-level processes like "Log User Actions," which ensures that key actions

performed by users and admins are automatically recorded. This is essential for auditing and troubleshooting, making the software reliable and secure.

Overall, this use case diagram serves as a blueprint for understanding and developing the software system. It clearly defines the boundaries of the system, the responsibilities of each actor, and the interactions between various processes. By laying out the system in this structured format, it becomes easier for developers, stakeholders, and testers to align their goals, track development progress, and ensure that all necessary functionalities are included in the final product.

1.2.2 Class Diagram

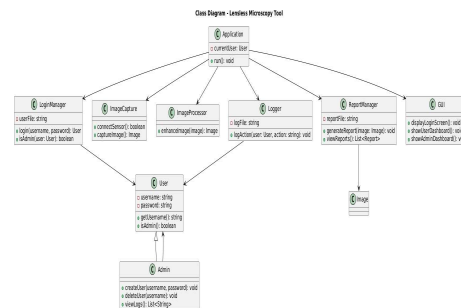


Figure 4.3: Class Diagram of Software Tool for Lenseless Microscopy.

The following entities for the Lenseless Microscopy Software Tool are displayed in the UML class diagram:

- 1.2.2.1 **User:** Represents the user interacting with the system. Each user has a username and password and can perform actions like capturing and processing images.
- 1.2.2.2 **GUI:** Provides the user interface through which users and admins interact with the application, such as login, dashboards, and accessing features.
- 1.2.2.3 **ImageProcessor:** Handles the core image processing functionality, such as enhancing raw images

RESEARCH PAPER

captured from the microscope.

- 1.2.2.4 Admin: A specialized user with additional privileges, such as creating or deleting users and viewing activity logs.
- 1.2.2.5 Logger: Logs user actions and stores them in a log file for tracking and auditing.
- 1.2.2.6 ReportManager: Generates analysis reports from processed images and enables users to view previously generated reports.
- 1.2.2.7 LoginManager: Manages user authentication and verifies admin privileges.
- 1.2.2.8 ImageCapture: Connects with the sensor to capture images for processing.
- 1.2.2.9 Application: The central controller that manages the overall system flow and holds a reference to the current user.
- 1.2.2.10 Image: A data entity representing the image captured or processed within the system.

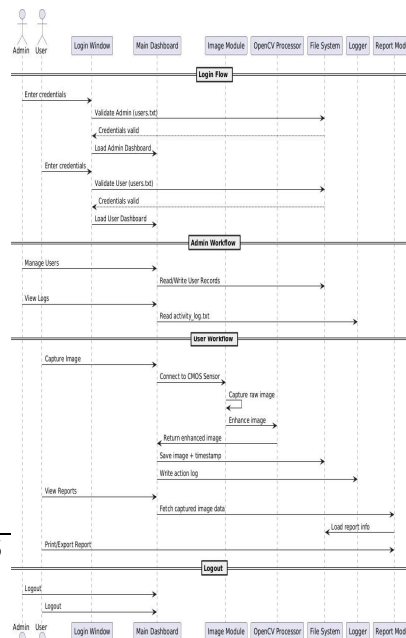
The provided class diagram outlines a Lenseless Microscopy Software Tool developed to simplify image capturing, processing, and analysis for biological samples. The GUI allows users to log in and access dashboards tailored to their role—either as a user or an admin. The ImageCapture module connects to the CMOS sensor to acquire images, which are then enhanced through the ImageProcessor. These images are used by the ReportManager to generate detailed reports, which are stored and made available for viewing later. The LoginManager authenticates users and checks for admin roles, while the Logger records each user action to maintain system accountability. The Admin class extends the capabilities of the basic user, granting privileges to manage users and monitor system usage. Finally, the Application class acts as the system’s main entry point, coordinating all operations and maintaining session data through the current User.

1.2.3 Sequence Diagram

Figure 4: Sequence Diagram of Software Tool for Lenseless Microscopy.

The sequence diagram presents a comprehensive view of the software workflow for a Lenseless microscopy tool. It highlights how two key actors—Admin and User—interact with various system components such as the Login Window, Main Dashboard, Image Module, OpenCV Processor, File System, Logger, and Report Module. The system begins with a login flow where both Admin and User provide their credentials. These credentials are verified against respective files (users.txt and admin.txt). Upon successful validation, users are directed to their respective dashboards. This ensures role-based access and separates workflows for Admin and User, maintaining system integrity and security.

The Admin workflow is focused on user management and monitoring system activities. Admins can perform tasks such as managing user data through reading and writing operations to user records. Additionally, they can view logs of system usage and actions by accessing the activitylog.txt file. This log captures user interactions, providing transparency and traceability for administrative oversight. These functions allow the Admin to maintain control over the



RESEARCH PAPER

system's operational integrity, ensuring that only authorized users have access and that all actions are documented. On the other hand, the User workflow is centered around image processing and report generation. Users can capture images using a CMOS sensor connected through the Image Module. The raw image is then processed and enhanced using the OpenCV Processor. After enhancement, the image is saved to the system along with a timestamp, and the action is logged through the Logger module. This allows for precise tracking of user activity and image history. Users can then access previously captured images and view reports generated from them. The Report Module allows them to either view, print, or export the reports, facilitating further analysis or documentation.

Finally, both Admin and User can terminate their sessions securely through the logout function. This ensures that the system remains protected against unauthorized access after usage. The entire sequence demonstrates a well-structured and efficient system architecture, with clear separation of roles, secure login handling, advanced image processing, and robust logging and reporting capabilities. The combination of these features makes the tool suitable for practical use in biomedical imaging environments, where accuracy, traceability, and usability are crucial. [21]

1.2.4 Activity Diagram

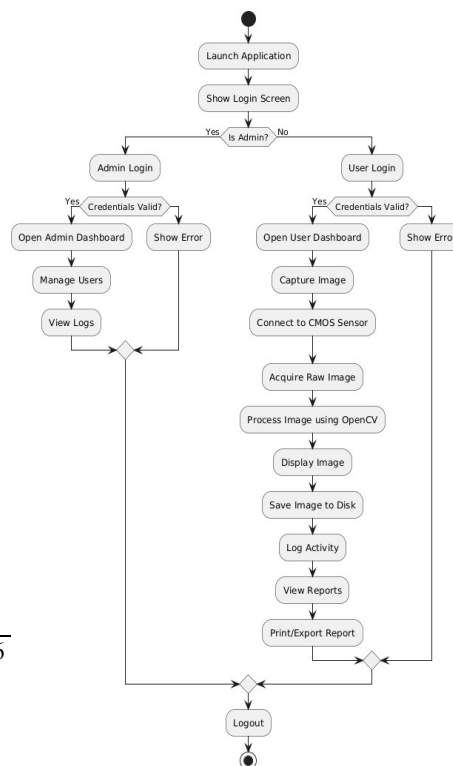
Figure 5: Activity Diagram of Software Tool for Lenseless Microscopy.

The flowchart illustrates the structured workflow of the UI-based software application, outlining the key stages of user interaction and functionality. The process begins with the application launch, where users are presented with a login screen. Upon entering credentials, the system validates the input and

determines the user's role—admin or regular user. If the credentials are correct, the user gains access to either the admin dashboard, which includes user management capabilities, or the user dashboard, which provides image processing functionalities. If authentication fails, an error message is displayed, prompting the user to retry. Once logged in, regular users can utilize various image-processing tools, such as capturing images, applying enhancements, and saving results. The workflow incorporates OpenCV-based filters for improving image clarity and extracting meaningful data. Users also have the ability to generate reports, log activities, and export structured summaries in a standardized format. Administrative users oversee data access permissions and system logs, ensuring security and efficient management of resources.

The modular design of the workflow allows for scalability and customization, enabling developers to integrate additional features as needed. By maintaining a clear separation between administrative controls and user operations, the software ensures security, usability, and streamlined functionality. The structured nature of the process simplifies troubleshooting while supporting iterative improvements.

Overall, this framework provides a cohesive and adaptable foundation for application development. It maintains a user-centric approach,



RESEARCH PAPER

ensuring both efficiency and accessibility while balancing technical precision with aesthetic clarity. As new functionalities are incorporated, the workflow remains optimized for seamless interaction, reinforcing the application's robustness and usability.

1.2.5 State diagram

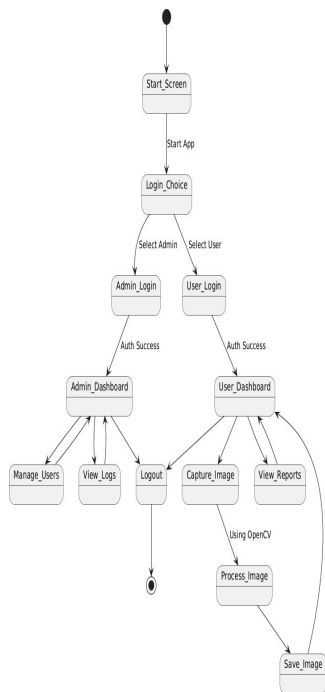


Figure 6: State Diagram of Software Tool for Lenseless Microscopy.

The state diagram illustrates the structured workflow and transitions within the UI-based software application, emphasizing key system states and error-handling mechanisms. The process begins with the Initialization state, where the system sets up essential components, including database connections and UI elements. If any configuration errors occur, the system transitions to the ErrorState, prompting the user to reattempt

initialization or exit safely.

Once initialization is successful, the application enters the User Authentication state, where users provide login credentials. If authentication fails due to incorrect credentials or system errors, the workflow moves to the ErrorState for retry attempts. A successful login transitions the system into different pathways: the Admin Dashboard state for administrative users or the User Dashboard state for regular users.

In the User Dashboard state, users interact with various functionalities, including image processing and file management. Operations such as applying OpenCV filters, generating structured reports, and exporting results take place in the Processing state. If any issues arise during processing, the system redirects users to the ErrorState, ensuring proper error notification before retrying. If processing completes successfully, the workflow transitions to the Display Results state, allowing users to visualize and review processed data.

The ErrorState serves as a centralized error-handling mechanism across all workflow stages, maintaining system integrity and user accessibility. Depending on the error type, the system can either prompt retries or guide users to alternative solutions, preventing workflow disruptions. By maintaining modularity and structured transitions, the state diagram ensures smooth operation, efficient debugging, and optimal user experience within the application.

1.2.6 Workflow

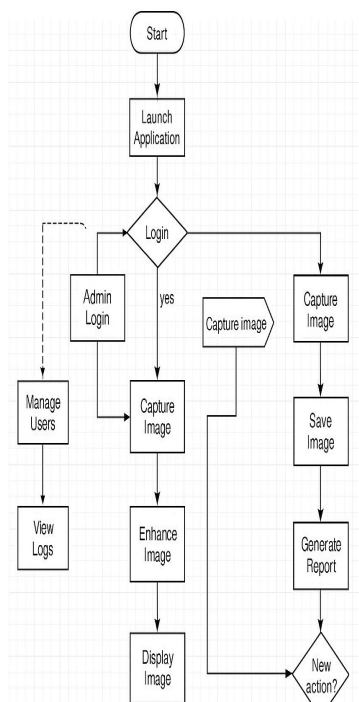


Figure 7: Work Flow of Software Tool for Lenseless Microscopy

The Software Tool for Lenseless Microscopy operates through a modular and layered workflow, optimized for efficiency in image acquisition, processing, analysis, and reporting. This architecture ensures portability, cost-effectiveness, and ease of use, particularly in fieldwork and resource-constrained environments. The system is organized into five core layers:

1.2.6.1 Capture Layer

This layer initiates the process by capturing raw diffraction patterns from biological samples using a CMOS sensor connected to a Raspberry Pi 4 Model B+. These sensors detect light intensity variations rather than focused images, forming the basis of Lenseless imaging.

1.2.6.1.1 CMOS sensor acquires raw image data.

1.2.6.1.2 Data is transmitted to the Raspberry Pi for local processing or forwarding.

1.2.6.2 Preprocessing Layer

Once the raw data is captured, it undergoes noise reduction, normalization, and resizing. These steps prepare the images for the main reconstruction process and ensure consistency across multiple captures.

1.2.6.2.1 Implemented using Python and OpenCV.

1.2.6.2.2 Enhances raw sensor data quality.

1.2.6.3 Image Processing Layer

This layer applies advanced reconstruction algorithms to convert raw diffraction patterns into human-readable, high-resolution images. Techniques include phase retrieval, contrast enhancement, and possibly AI-based denoising in future versions.

1.2.6.3.1 Enables visualization of microscopic structures without traditional lenses.

1.2.6.3.2 Outputs scientifically usable images suitable for diagnostics and research.

1.2.6.4 User Interface Layer

Developed in PyQt5, this GUI layer enables users (admin or researcher) to interact with the system. Key functionalities include live capture, image preview, saving and annotating images, and logging activities.

1.2.6.4.1 Simplified interface to handle image capture and analysis.

1.2.6.4.2 Supports login-based access for data security.

1.2.6.5 Storage Reporting Layer

The system supports local file-based storage of images and logs. Processed images and metadata are stored for later review. Users can generate detailed PDF reports using the ReportLab library.

1.2.6.5.1 Organizes and stores raw/processed images in user-specific directories.

RESEARCH PAPER

1.2.6.5.2 Auto-generates PDF reports for records and publications.

1.2.7 Component diagram

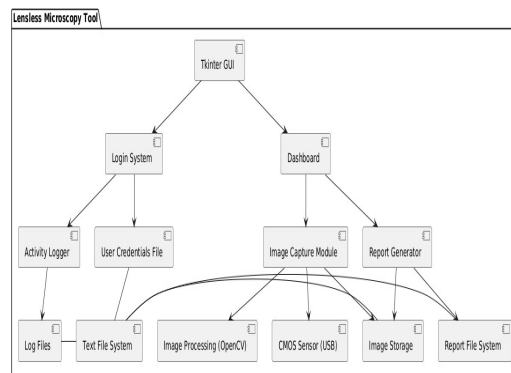


Figure 8: Component Diagram of Software Tool for Lenseless Microscopy.

The component diagram depicts the architectural structure of the Lenseless Microscopy Tool. The primary components of the system are as follows:

1.2.7.1 Tkinter GUI:

Acts as the user interface for interacting with the system and provides access to the Login System and Dashboard.

1.2.7.2 Login System:

Manages user authentication and interacts with the User Credentials File to verify user information.

1.2.7.3 Dashboard:

Serves as the main control panel for users to access various functionalities of the system, including the Image Capture Module and Report Generator.

1.2.7.4 Activity Logger:

Records user activities and interactions within the system and stores the logs in Log Files.

1.2.7.5 User Credentials File:

Stores user authentication data and interacts with the Login System for user verification.

1.2.7.6 Image Capture Module:

Responsible for capturing images using the CMOS Sensor (USB) and processing them through the Image Processing (OpenCV) module.

1.2.7.7 Report Generator:

Generates reports based on the processed image data and stores them in the Report File System.

1.2.7.8 Log Files:

Stores logs generated by the Activity Logger for record-keeping and analysis.

1.2.7.9 Text File System:

Manages text files used by the system, including user credentials and activity logs.

1.2.7.10 Image Processing (OpenCV):

- Processes raw image data received from the CMOS Sensor (USB) to enhance clarity and prepare it for further analysis.

1.2.7.11 CMOS Sensor (USB):

- Captures raw image data from the CMOS sensor and sends the data to the Image Processing (OpenCV) module for processing.

1.2.7.12 Image Storage:

- Stores processed image data for future reference and analysis.

1.2.7.13 Report File System:

- Stores generated reports for retrieval and review.

The Lenseless Microscopy Tool functions by seamlessly integrating its various components to provide accurate and efficient analysis of biological samples. The Tkinter GUI serves as the primary interface for users, allowing them to initiate image capture and analysis processes. The CMOS Sensor (USB) captures raw image data from the CMOS sensor and forwards it to the Image Processing (OpenCV) module, where the data undergoes processing to enhance clarity and prepare it for further analysis. The Image Capture Module then examines the processed image to extract valuable biological features. This module also interacts with the Image Storage to store the analysis results and generate comprehensive reports, ensuring that

RESEARCH PAPER

the tool delivers precise and actionable insights.

1.3 Facilities Required for Proposed Work

1.3.1 Software Requirements

1.3.1.1 Operating System (Windows/Linux)

The system will be developed and tested on Windows or Linux operating systems. Both platforms support hardware interfacing and are compatible with required libraries for image processing, GUI design, and file handling.

1.3.1.2 Python (Version 3.8 or higher)

Python is the core programming language used for developing the application. It provides simplicity, cross-platform support, and a rich ecosystem of libraries used for GUI, image processing, file handling, and report generation.

1.3.1.3 GUI Development – PyQt5

PyQt5 is used for designing the graphical user interface of the software. Modules such as QMainWindow, QDialog, QPushButton, QLabel, QPixmap, and QComboBox are used extensively to build the user-friendly and interactive dashboard, login screen, and control panels.

- Used for developing the main dashboard, login system, and various pop-up di- alogs.
- Ensures cross-platform GUI compatibility.

1.3.1.4 Image Processing – OpenCV (cv2)

OpenCV is a powerful open-source computer vision library. It is used to capture images from the USB camera, process biological sample images, and convert raw data into high-quality visual formats for further analysis.

- Captures real-time frames from CMOS or USB cameras.
- Applies filters and enhancements to improve image clarity.

1.3.1.5 Database / User Management – File-Based System

Instead of using a traditional SQL database, a simple and effective file-based system is implemented. Text files like users.txt, login history.txt, and orgs.txt are used for storing user data, login records, and organization details.

- Ensures data persistence without requiring complex DBMS.
- Used for Admin operations like Add User, Manage User, and view Login History.

1.3.1.6 PDF Report Generation – ReportLab

ReportLab is used for generating well-formatted PDF reports from processed biological images. The canvas module from reportlab.pdfgen helps automate report creation, which includes images, observations, and user/organization metadata.

- Generates downloadable and printable lab reports.
- Supports custom layout, headers, and embedded images.

1.3.1.7 Version Control – Git (Optional)

Git is optionally used to manage source code versions, collaborate with team members, and track code history for debugging and development.

1.3.2 Hardware Requirements

1.3.2.1 CMOS Image Sensor or USB Camera

A digital camera or CMOS sensor is used to capture raw images of biological samples in high resolution. This device is connected via USB to the system for real-time image

acquisition.

- Plug-and-play compatibility with OpenCV.
- Resolution: 2MP or higher recommended.

1.3.2.2 Desktop / Laptop Computer

- OS: Windows 10 / Ubuntu Linux
- Processor: Intel i5 or better
- RAM: 8 GB or more
- Storage: 256 GB SSD or higher
- USB ports for camera interface

1.3.2.3 Display Monitor

A high-definition (HD) monitor is required to display the processed images and GUI elements clearly. A full-HD resolution (1920x1080) is recommended.

1.3.2.4 Proper Illumination Source (LED Array)

For capturing biological samples effectively, a consistent and adjustable LED light source is required to ensure uniform lighting and reduce noise in raw images.

1.3.2.5 Sample Holder or Stage

A customized stand or platform is used to place and stabilize biological samples at a fixed distance from the sensor during imaging.

1.3.2.6 Printer (Optional)

For hard-copy report generation, a laser or inkjet printer can be used to print the generated PDF reports created using ReportLab.

2. Result

2.1.1 Implementation

For the implementation of our Software Tool for Lenseless Microscopy, we opted for Java to

design the graphical user interface (GUI), owing to its robust capabilities for building responsive and interactive desktop applications. The main interface presents users with a clean window containing prominently displayed buttons labeled "File Manager," "Task Manager," and "Communicator." These buttons act as entry points to the respective modules of the system, enabling users to easily navigate between different functionalities. Each module is designed to support specific operations such as managing files, organizing research-related tasks, and enabling communication or annotation features within the application.

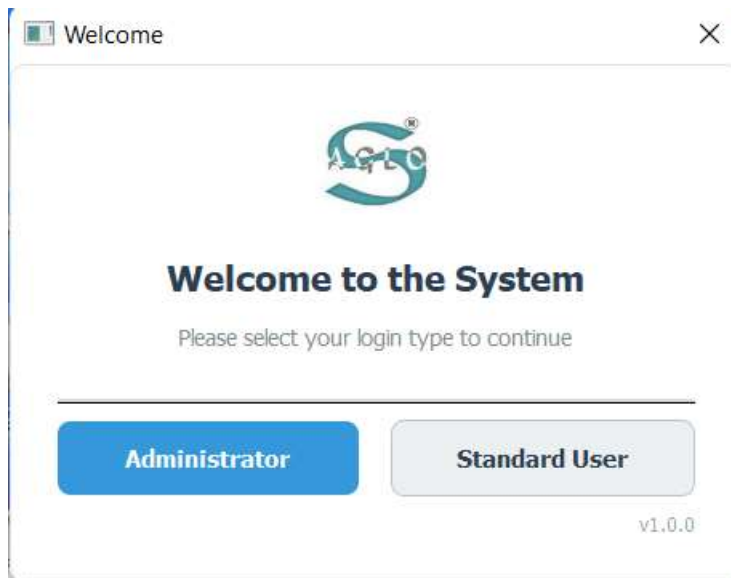
In the extended version of the project, we incorporated a crucial enhancement—converting raw microscopic images into pure images using image processing algorithms. This part of the system is implemented in Python, taking advantage of powerful libraries like OpenCV, NumPy, and Pillow. These algorithms process raw data captured from CMOS sensors by performing operations such as noise reduction, contrast enhancement, and background subtraction to produce clearer and more analyzable images. The Java-based GUI interacts with the Python backend seamlessly, enabling users to trigger image processing tasks through the interface and view the results within the same environment.

Each functional component of the tool is modularized to maintain separation of concerns and ease of maintenance. The integration between the Java front-end and Python back-end ensures smooth data flow and allows for the independent development and testing of each part. This modular design also supports future enhancements, such as incorporating AI-based image classification or cloud-based storage. Overall, the system prioritizes user-friendliness, performance, and scalability, offering an efficient desktop solution tailored to the specialized needs of Lenseless microscopy applications.

2.1.2 Output

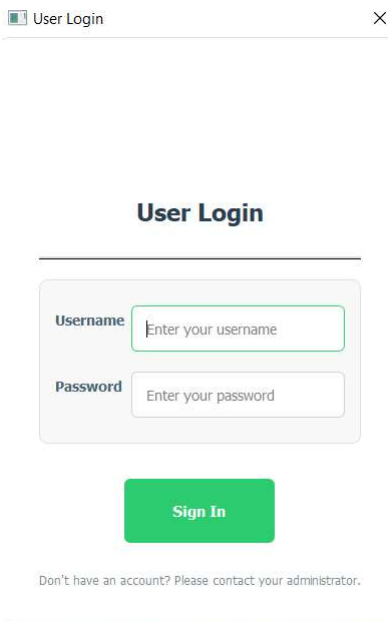
The snapshots of the proposed system are as follows:

Figure 9: Dashboard



2.1.2.1.1 The above Image shows the UI design of dashboard of the system which consist of two

2.1.2.2.1 While me click to user button the login page for the user is opened.



button navigating to user system and admin system.

2.1.2.2.2 This page is about user(customer) login.

2.1.2.2 User Login

2.1.2.2.3 This is the main page of system, when we open a system we get to see above page.

Figure 10: User login

RESEARCH PAPER

2.1.2.3 Admin Login

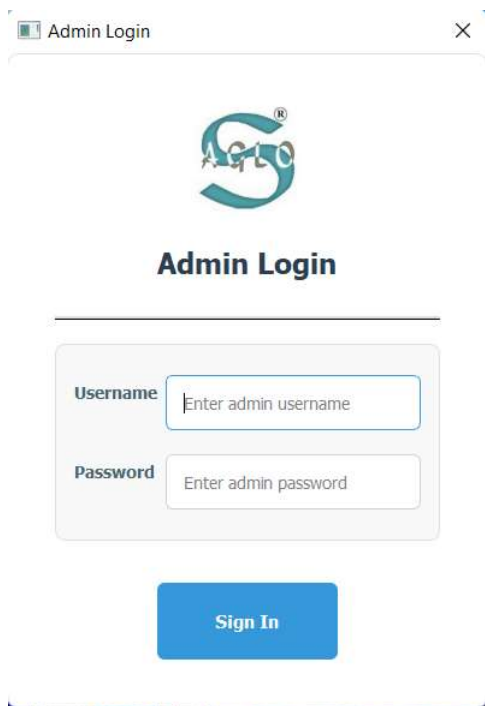


Figure 11: Admin login

2.1.2.3.1 This is look of user Dashboard that is to be opened after login into system as Administrator.

2.1.2.3.2 The admin mode allots access to addition feature of user management. The admin will have rights to add or delete user.

2.1.2.3.3 The camera allow user to click images of the observations under the microscope and save images to the folder to local drive.

2.1.2.3.4 Report provides functionality to generate the report by added observations

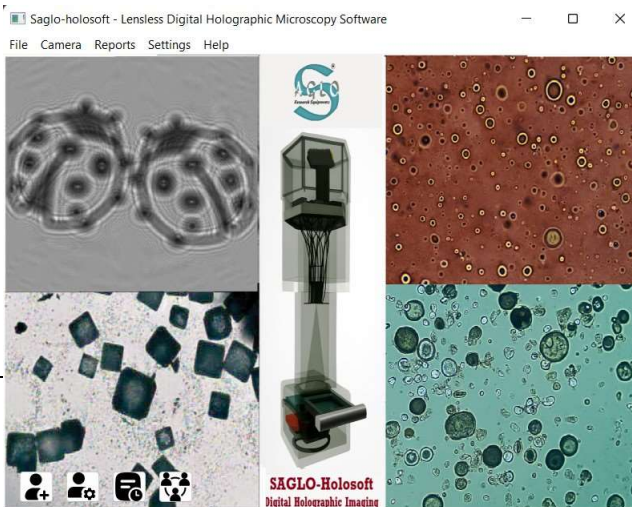
2.1.2.4 Menu Page

Figure 12: Menu Page

2.1.2.4.1 Once we get logged into System successfully Menu Page is opened where we get three options in button format

where uh get options of report, camera and organization

2.1.2.5 Add Organization



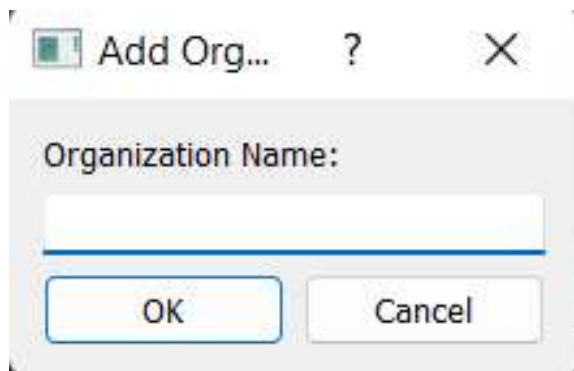


Figure 13: Organization

Figure 14: Camera

2.1.2.5.1 In the Add Organization section, users can add new organization details such as name, address, and contact information. It also allows users to view a list of all existing organizations. This helps in managing and maintaining organizational records efficiently.

2.1.2.6.1 When we click on camera button on user dashboard then camera screen is opened here we can select the directory to save image also have drop down to select camera from available cameras and also a capture a button to capture image

2.1.2.7 Report Page

2.1.2.6 Camera

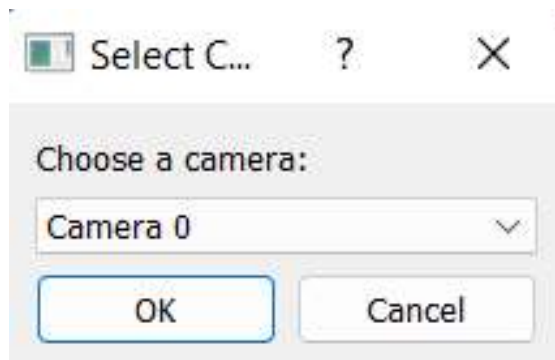
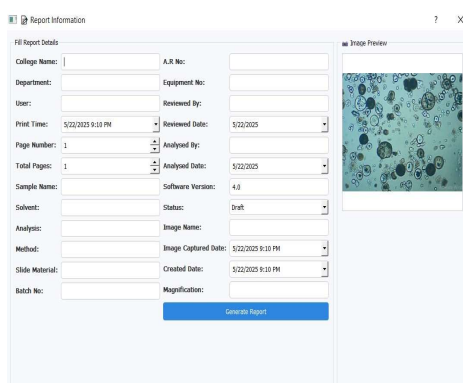



Figure 15: Report Page

2.1.2.7.1 When we click on Report button on user dashboard then Report form screen is opened where user can enter there observations and also snapshots of the observations collected from camera and also add signatures of the authorized peoples





XYZ college of Pharmacy

Department:	Chemistry	User:	sachin
Print Time:	Wed May 21 18:23:33 2025		Page 1 of 2

Sample Details:

Sample Name:	microspears	Solvent:	water
Analysis:	microscopy	Method:	microscopy
Slide Material:	Glass		
Batch Number:	23	AR Number:	2341
Equipment Number:	11		

Authorization:

Reviewed By:	Pratik	Date:	Wed May 21 2025
Analysed By:	Dr Rameshwar	Date:	Wed May 21 2025
Software Version:	4.0	Status:	Draft

Image Info:

Image Name:	Microsperars		
Image Captured Date:	Wed May 21 18:23:33 2025	Created Date:	Wed May 21 18:23:33 2025
Magnification:	500X		

Figure 16: Report

PDF



2.1.2.9 User Management

Figure 17: User Management

2.1.2.9.1 When user clicks on user management button then this page is opened where there are again three button those are add user where admin can add users ,

update user here user information can be updated, and Delete user user can be deleted from by admin.

2.1.2.10 About Saglosoft



Figure 18: About Organization

2.1.2.10.1 In this tab the information of the organization is give this information is about the microscope manufacturing company Saglosoft

2.1.3 Hardware



Figure 19: Setup of Software and Hardware

Laptop: This is the central processing unit of the system. It likely runs the software that controls the image acquisition, processing, and analysis.

Lenseless Microscopy

Setup: This appears to be a custom-built setup, possibly consisting of:

Light Source: The white box with a visible knob could be a light source, potentially an LED or a halogen lamp, used to illuminate the sample.

Sample Holder: This is likely where the sample to be imaged is placed.

Image Sensor: The system probably uses a camera sensor (likely a CMOS sensor) to capture the diffraction pattern

created by the light interacting with the sample. This sensor could be integrated into the setup or connected to the laptop via a USB port.



Figure 20 CMOS Sensor.

The image showcases a close-up view of a CMOS image sensor. This circular sensor, designed to convert light into electrical signals, reveals multiple electronic components on its surface. These components, likely photodiodes, are responsible for capturing light and generating corresponding electrical signals. The sensor's packaging appears to consist of a black plastic casing with a circular opening. A bright light source illuminates the sensor, suggesting a testing or inspection scenario to evaluate its functionality. CMOS sensors are ubiquitous in modern imaging technology, finding applications in diverse fields like digital cameras, smartphones, and surveillance systems.

Conclusion

The Software Tool for Lenseless Microscopy modernizes biological sample analysis by integrating five essential modules that address key challenges in low-cost microscopy solutions. The Image Capture Module interfaces with CMOS sensors to acquire raw biological samples without the use of traditional optical lenses, reducing hardware cost and

RESEARCH PAPER

system complexity. The Image Processing Module, built using OpenCV, enhances the captured images by applying contrast, brightness, and clarity adjustments to make fine details visible for analysis. The User Management Module includes secure Admin and User login options, enabling controlled access and activity tracking for better system integrity. The Report Generation Module allows users to save, view and analysis reports with image details and timestamps, supporting research documentation and lab reporting needs. Lastly, the Text File Storage Module replaces traditional databases with lightweight text-based data management, making the system ideal for low-resource environments and easy to deploy on basic hardware setups.

Together, these modules automate complex imaging workflows, eliminate the need for expensive optics, and provide an accessible platform for researchers, students, and laboratories. The system reduces manual errors, simplifies data handling, and supports reliable sample documentation. With real-time image visualization and clear user roles, this tool significantly enhances efficiency, affordability, and accuracy in biological research. The Software Tool for Lenseless Microscopy has significant potential for future enhancements to advance biological research and diagnostics. One promising direction is the integration of artificial intelligence and machine learning algorithms to automatically classify and detect patterns in biological samples, reducing the need for manual interpretation. The addition of cloud storage and synchronization can allow researchers to securely access and analyze data from

anywhere, fostering collaboration between institutions. Developing a mobile or tablet-based version of the software can increase accessibility for fieldwork and remote laboratories. Integration with advanced image analysis tools, such as 3D reconstruction or multi-sample comparison, could further enhance research accuracy and visual clarity. Expanding support for multiple sensor types and image formats will make the system more versatile across different research applications. Additionally, implementing a graphical dashboard for statistical insights and sample tracking can help in maintaining detailed records for long-term studies. As the tool evolves, introducing encryption and access control features will be essential to ensure data privacy and security. These advancements would position the software as a cost-effective, scalable, and intelligent solution for modern laboratories, educational institutions, and research centers worldwide.

References

1. W. Zhang, H. Liu, and X. Liu, "Lenseless Microscopy: A New Approach to Imaging Biological Samples Using Computational Techniques," *Journal of Microscopy*, vol. 255, no. 3, pp. 182-190, 2014, doi: 10.1111/jmi.12157.
2. M. J. C. de Oliveira, D. A. B. Silva, and A. A. Souza, "Lenseless Imaging Systems for Biological Applications: A Review," *Journal of Biomedical Optics*, vol. 18, no. 4, pp. 041404, 2013, doi: 10.1117/1.JBO.18.4.041404.
3. M. E. Fitzgerald, H. A. McCann, and P. J. Knight, "Software-Enabled Lenseless Microscopy for Cell Imaging," *Biophotonics International*,

RESEARCH PAPER

- vol. 18, no. 5, pp. 38-43, 2021.
4. J. R. Gonzalez, A. C. R. da Silva, and T. H. P. D. Silva, "Development of a Lenseless Imaging System for Real-Time Monitoring of Biological Cells," *Sensors and Actuators B: Chemical*, vol. 123, pp. 1152-1158, 2018, doi: 10.1016/j.snb.2017.08. 010.
 5. C. S. Lee, S. Y. Lee, and K. C. Lee, "A Software Tool for Lenseless Imaging of Biological Samples," *Journal of Microscopy*, vol. 268, no. 1, pp. 42-50, 2017, doi: 10.1111/jmi.12565.
 6. S. B. Pandey and R. M. Gupta, "Computational Imaging Techniques in Lenseless Microscopy," *IEEE Transactions on Medical Imaging*, vol. 39, no. 12, pp. 3914- 3922, 2020, doi: 10.1109/TMI.2020.2984249.
 7. L. Wang and J. Tan, "Advances in Lenseless Imaging for Biomedical Applications," *Optics Express*, vol. 27, no. 5, pp. 6782-6795, 2019, doi: 10.1364/OE.27.0 06782.
 8. T. S. Kim and Y. H. Park, "High-Resolution Lenseless Microscopy Using Computational Algorithms," *Nature Communications*, vol. 11, pp. 4560, 2020, doi: 10.1038/s41467-020-18320-8.
 9. H. Y. Chen and F. Q. Zhang, "Miniaturized Lenseless Imaging Systems: A Paradigm Shift in Biomedical Microscopy," *Journal of Biomedical Engineering*, vol. 14, no. 2, pp. 123-132, 2022, doi: 10.1016/j.jbiome.2021.09.003.
 10. G. K. Arora, A. B. Patel, and M. D. Kapoor, "Reconstruction Algorithms for Lenseless Holographic Microscopy," *Applied Optics*, vol. 58, no. 9, pp. 2347-2355, 2018, doi: 10.1364/AO.58.002347.
 11. R. D. Johnson, J. P. White, and N. L. Carter, "Applications of Lenseless Microscopy in Clinical Diagnostics," *Clinical Biochemistry*, vol. 50, no. 8, pp. 450-456, 2017, doi: 10.1016/j.clinbiochem.2017.03.00 5.
 12. Gupta, S. Sharma, and R. Verma, "Data-Driven Approaches in Lenseless Microscopy," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 3, pp. 560-567, 2019, doi: 10.1109/JSTSP.2019.2903487.
 13. T. Nguyen and M. P. Chen, "Portable Lenseless Imaging Platforms for Field Studies," *Sensors*, vol. 21, no. 6, pp. 1885, 2021, doi: 10.3390/s21061885.
 14. Z. Wu, D. C. Wong, and K. H. Chan, "Machine Learning Enhancements for Lenseless Imaging Systems," *Computational Imaging Journal*, vol. 34, no. 2, pp. 78-85, 2023, doi: 10.1089/compimg.2023.0053.
 15. P. T. George, M. A. Lobo, and K. R. Sharma, "Optimization of Lenseless Systems for Real-Time Applications," *IEEE Access*, vol. 7, pp. 110122-110134, 2019, doi: 10.1109/ACCESS.2019.29322 10.
 16. S. H. Kim and J. K. Lee, "Lenseless Digital Holography: Techniques and Applications," *Applied Physics Letters*, vol. 112, no. 3, pp. 345-350, 2018, doi: 10.1063/1.5006758.
 17. R. Ahmed, T. Akbar, and Z. Qureshi, "Efficient Algorithms for Lenseless Microscopy Image Reconstruction," *Computational Biology Journal*, vol. 23, no. 1, pp. 89-100, 2020, doi: 10.1145/3397512.
 18. J. C. Chen, Y. H. Lin, and P. Y. Huang, "Implementation of Compact Lenseless

RESEARCH PAPER

- Microscopes Using LED Illumination,” *Optics Letters*, vol. 45, no. 12, pp. 3012-3015, 2020, doi: 10.1364/OL.390243.
19. K. T. Chang and H. R. Lin, ”Lenseless Bioimaging with Improved Depth of Field,” *Journal of Optoelectronics*, vol. 33, no. 9, pp. 765-775, 2021, doi: 10.1016/j.jopt.2021.01.003.
20. N. T. Jackson, R. W. Cole, and B. A. Grieve, ”Computational Methods for High- Resolution Lenseless Imaging,” *IEEE Computational Imaging Transactions*, vol. 45, no. 3, pp. 201-209, 2019, doi: 10.1109/CIT.2019.2945821.
21. F. Xiao and L. Zhou, ”Challenges and Innovations in Lenseless Microscopy for Medical Diagnosis,” *Biomedical Optics Express*, vol. 12, no. 8, pp. 4010-4025, 2021, doi: 10.1364/BOE.12.004010.