

# Microbe Detection Using Image Registration and CNN

Suraj Kumar<sup>1</sup>, Mogalapu Jaya Srikar<sup>2</sup>, and Dr. Anil Kumar Kakelli<sup>3</sup>

<sup>1</sup>Computer Science and Engineering, VIT University, Vellore

<sup>2</sup>Computer Science and Engineering, VIT University, Vellore

<sup>3</sup>Associate Professor, School of Computer Science and Engineering, VIT University, Vellore

**Abstract** This research presents a pioneering approach by integrating image registration techniques with advanced CNNs using transfer learning, elevating precision in classifying 'Healthy,' 'Brain Tumour,' 'Breast Cancer,' and 'Lung Cancer' in medical images. The methodology evolves the CNN from a foundational architecture through augmented layers, incorporating batch normalization and early stopping for enhanced discernment. Transfer learning models (VGG16, ResNet50, Inception, and Xception) further refine the CNN's capabilities. Concurrently, image registration algorithms (chi-squared shift, cross-correlation-based shift, optical flow-based shift, and an innovative pixel-wise average shift) align and register images for consistent analysis, ensuring standardized inputs. This synergy avoids redundancy and enhances microbial detection accuracy. Encapsulated in a Flask framework, the system enables users to upload MRI images for real-time cancer classification, contributing to automated microbe detection and diagnosis. In practical scenarios, our algorithmic approach addresses the challenges of manual disease detection, presenting significant advancements in medical imaging, computer vision, and healthcare technology.

**Keywords:** *Image Registration, Geometrically, Superimposing, Biomedical Images, CNN, Transfer Learning*

## 1 Introduction

The landscape of medical imaging and microbial detection stands at the threshold of a transformative paradigm, as outlined in this research endeavor. In our pursuit of precision and efficacy, we introduce a novel fusion of image registration techniques and advanced convolutional neural networks (CNNs) employing transfer learning. This groundbreaking approach targets the enhancement of accuracy in medical image analysis, specifically honing in on the classification of 'Healthy,' 'Brain Tumour,' 'Breast Cancer,' and 'Lung Cancer.'

Central to our methodology is the strategic synthesis of image registration techniques and CNNs, offering a nuanced departure from conventional practices. Commencing with a foundational CNN architecture, our model undergoes a meticulous evolution marked by augmented convolution and max-pooling layers. The culmination of this evolution introduces sophisticated features—batch normalization and early stopping—refining the CNN's ability to discern intricate patterns critical for precise microbial classification.

Distinguishing our methodology is the incorporation of transfer learning models—VGG16, ResNet50, Inception, and Xception—each contributing to the CNN's heightened acuity in classifying medical images. This strategic amalgamation propels our approach beyond traditional boundaries, acknowl-

edging the nuanced intricacies of disease classification.

we introduce a suite of image registration algorithms—chi-squared shift, cross-correlation-based shift, optical flow-based shift, and an innovative pixel-wise average shift. These algorithms collectively align and register input images, ensuring accurate and consistent analysis. The pivotal role played by these algorithms lies in their ability to normalize input images, establishing a steadfast foundation for subsequent CNN analysis.

The synchronization of image registration and CNNs is an orchestration of meticulous precision, capitalizing on the individual strengths of each methodology. This seamless collaboration ensures that registered images serve as optimal inputs for subsequent CNN-based classification, circumventing redundancy and fostering a synergetic enhancement of microbial detection accuracy.

Manual detection of microbial patterns in medical images introduces inherent challenges, including subjectivity, variability, and the potential for human error. The limitations of human perception, particularly in discerning subtle or complex patterns, compromise the accuracy of disease classification and prolong the diagnostic timeline. In contrast, the integration of Convolutional Neural Networks (CNNs) and transfer learning models presents a paradigm shift towards automated detection, overcoming the drawbacks of manual methods.

CNNs excel in automated pattern recognition, leveraging hierarchical feature extraction for intricate detail discernment. This ensures consistency and reproducibility, mitigating the variability introduced by manual detection across different practitioners and settings. The efficiency of automated detection with CNNs significantly reduces analysis time, facilitating timely diagnoses crucial for improved patient outcomes.

The upgradation of CNNs further enhances accuracy, incorporating increased convolution and max-pooling layers for a broader receptive field. Batch normalization optimizes the training process by stabilizing input for each layer, while early stopping prevents overfitting, ensuring model generalization to new data. This strategic combination refines the CNN's ability to discern complex features crucial for precise microbial classification, contributing to heightened accuracy. The proposed system's functionality, encapsulated within a Flask framework, underscores our commitment to a seamless integration of the frontend and model weights. This

integrated interface empowers users to upload MRI images, enabling real-time cancer classification and delivering crucial diagnostic insights into 'Healthy', 'Brain Tumour', 'Breast Cancer' and 'Lung Cancer'.

In scenarios where manual disease detection proves challenging, our algorithmic approach emerges as a viable solution, marking a significant contribution to the realms of medical imaging, computer vision, and healthcare technology. As we embark on a detailed exploration of our proposed methodology, we anticipate a paradigm shift in automated microbial detection and diagnosis, setting new benchmarks in technological innovation and medical research.

For the flow chart of Image Registration refer [Figure 1](#).

## 2 Problem Statement

Current methods for microbial detection and classification in medical imaging face challenges in achieving both accuracy and efficiency. Traditional approaches lack the necessary precision to discern subtle differences in image patterns, particularly in the classification of 'Healthy,' 'Brain Tumour,' 'Breast Cancer,' and 'Lung Cancer.' Additionally, manual disease detection remains a formidable task in practical scenarios due to its inherent complexity.

This research addresses these challenges by proposing a novel paradigm that seamlessly integrates image registration techniques with advanced convolutional neural networks (CNNs) using transfer learning. The existing gap lies in the inability of conventional methods to provide consistent and standardized inputs for accurate microbial analysis. Image registration techniques, such as chi-squared shift, cross-correlation-based shift, optical flow-based shift, and an innovative pixel-wise average shift, are currently underutilized in tandem with advanced CNNs.

Our research seeks to strategically combine these methodologies to create a unified approach, beginning with a foundational CNN architecture that evolves through augmentation stages. The incorporation of increased convolution and max-pooling layers, alongside batch normalization and early stopping, aims to refine the CNN's ability to discern complex features essential for precise microbial classification.

The research further explores the use of transfer learning models, including VGG16, ResNet50, Inception, and Xception, to enhance the CNN's proficiency in classifying medical images. This combination of methodologies, if not addressed, results in redundant processes, limiting the potential for synergistic enhancement in microbial detection accuracy.

Our proposed solution encapsulates the functionality within a Flask framework, providing a seamless integration of the frontend and model weights. This integration aims to facilitate real-time cancer classification, offering crucial diagnostic insights into 'Healthy,' 'Brain Tumour,' 'Breast Cancer,' and 'Lung Cancer.' The significance of this research lies in the potential to revolutionize automated microbe detection and

diagnosis, contributing to advancements in medical imaging, computer vision, and healthcare technology.

## 3 Literature Review

### 1. An Efficient Microbes Detection System using Microscopic Images via Morphological and Correlation Based Features

The manual identification of algae in water bodies has long been recognized as a intensive and expert-dependent process. This research offers a solution by automating this identification process. Specifically, a hybrid algorithm combining pixel clustering and the Kirsch filter has been developed, demonstrating a high level of accuracy in extracting microbial bodies from images. Furthermore, the study extensively evaluates various classification methods for the automatic identification process. Comparison is made between Classification and Regression Trees (CART), K-nearest-neighborhood, Gaussian Naive Bayes, Linear Regression, Linear discriminant analysis, and Support vector classifier (SVC). The findings indicate that the CART algorithm emerges as the most stable and consistent performer, as evidenced by its high values in recall, precision, and accuracy [1]. The SVC algorithm also demonstrates commendable performance, securing the second position. The research presented in this paper addresses a crucial need in the field of computational biodiversity studies, with a focus on contributing to the preservation of ecological balance on Earth. By leveraging advancements in image processing and machine learning algorithms, the work proposes a pipeline of algorithms aimed at supporting biodiversity research.

### 2. High-Resolution Chemical Mapping and Microbial Identification of Rhizosphere using Correlative Microscopy

The presented approach offers a groundbreaking solution for the simultaneous investigation of soil-root-microbe interactions at micro-to nano-meter spatial resolutions [2]. By integrating a resin embedding and sectioning method tailored for *Zea mays* rhizosphere, along with an analytical workflow enabling the concurrent use of up to six complementary instruments/techniques, They have achieved a significant advancement in the field. Additionally, a robust system for data and image correlation was established. The utilization of hydrophilic, immunohistochemistry-compatible LR white resin for sample embedding, coupled with the implementation of waterjet cutting and the avoidance of surface polishing, ensured high-quality embedding without nanoscale smearing. Verification of embedding quality through Helium Ion Microscopy (HIM) further validated the effectiveness of this approach. Identification of bacteria

within the embedded soil was successfully accomplished through Catalyzed Reporter Deposition-Fluorescence In Situ Hybridization (CARD-FISH), mitigating potential interferences from auto fluorescence emitted by soil particles and organic matter. The chemical mapping of the rhizosphere was achieved through a comprehensive suite of techniques, including Scanning Electron Microscopy with Energy-dispersive X-ray analysis (SEM-EDX), Time-of-Flight Secondary Ion Mass Spectrometry (ToF-SIMS), nano-focused Secondary Ion mass Spectrometry (nanoSIMS), and confocal Raman spectroscopy (-Raman). By seamlessly integrating high-resolution correlative characterization across these six distinct techniques and implementing image registration, They have demonstrated the capability of this method to meet the rigorous demands of multi-modal characterization, enabling both chemical mapping of the rhizosphere and the identification of spatial bacterial organization. Ultimately, this novel approach serves as a versatile platform for integrating various 2D analytics, providing a comprehensive understanding of rhizosphere processes and their ecological significance. They anticipate that this method will pave the way for significant advancements in the study of soil-root-microbe interactions and their broader ecological implications.

### **3. A New Generation of the IMAGIC Image Processing System**

In conclusion, the IMAGIC-5 software system represents a significant advancement in modern microscopy, addressing the critical need to quantify complex biological, medical, and material phenomena in two, three, or even four dimensions. The design considerations of IMAGIC-5 are a testament to the rigorous requirements imposed by such data processing tasks. This comprehensive system encompasses a range of powerful features, including facilities for multivariate statistical analysis of large datasets, correlation averaging of two-dimensional crystals, and three-dimensional reconstruction of macromolecular structures. Importantly, IMAGIC-5 accommodates diverse molecular arrangements, whether as two-dimensional crystals, helices, or single particles with arbitrary pointgroup symmetry. A particularly noteworthy innovation of IMAGIC-5 is its novel angular reconstitution approach, enabling the swift determination of high-resolution three-dimensional structures of uncrystallized molecules. This breakthrough holds significant promise for advancing our understanding of molecular structures. The paper has provided an insightful overview of IMAGIC-5, addressing its general organization, user interaction strategy, file structure, and extendibility [3]. Practical examples have been presented to illustrate the application and versatility of the system. In summary, IMAGIC-5 stands as a powerful tool in the realm of modern microscopy, poised to make substantial contributions to the quantification and analysis of complex biological, medical, and material phenomena. Its innovative features and capabilities hold great potential for furthering

our knowledge and capabilities in the fields of biology, medicine, and material sciences.

### **4. Automated microorganisms activity detection on the early growth stage using artificial neural networks**

In conclusion, this paper introduces a pioneering approach utilizing a non-contact optical technique for the early assessment of microbial activity. The proposed method leverages the laser speckle contrast imaging technique in conjunction with artificial neural network (ANN) based image processing. The process of evaluating microbial activity involves capturing time-variable laser speckle patterns within a given sample, followed by ANN-based image processing and visualization of the obtained results. The technology aims to measure key parameters of microbial activity, such as growth speed, and subsequently apply these findings for the quantification of live microbes. Importantly, it is anticipated that this innovative approach will lead to a more efficient evaluation of colony forming units (CFU), potentially delivering results two to six times faster compared to the standard counting methods traditionally employed for CFU enumeration [4]. By offering a non-contact, rapid, and accurate means of assessing microbial activity, this technology holds significant promise for expediting microbial studies and applications across various domains, including biotechnology, medical research, and industrial processes. The proposed approach represents a substantial advancement in microbial analysis techniques and stands to benefit a wide range of scientific and industrial endeavors.

### **5. Image registration methods: A Survey**

In conclusion, this paper offers a comprehensive review of both recent and classic methods in image registration, a pivotal process in computer vision and image analysis. Image registration involves the alignment of multiple images of the same scene taken under different conditions, such as varying times, viewpoints, or sensors. Geometrically aligning these images is crucial for a wide range of applications. The reviewed approaches are systematically classified based on their nature, distinguishing between area-based and feature-based methods. Furthermore, they are categorized according to the four fundamental steps in the image registration procedure: feature detection, feature matching, mapping function design, and image transformation and resampling. Each method's key contributions, advantages, and limitations are discussed to provide a comprehensive understanding of their applicability and performance [5]. The paper also addresses pertinent challenges and problematic issues in image registration. Additionally, it offers insights into potential directions for future research, highlighting areas where further advancements and innovations are needed. The primary objective of this paper is to serve as a valuable and extensive reference for researchers engaged in image registration, regardless of their specific application domains. By consolidating and evaluating a diverse range of methods,

this review aims to facilitate informed decision-making and inspire future developments in this critical area of computer vision and image processing.

## **6. Image Registration Techniques: An Overview**

In conclusion, this paper underscores the critical significance of image registration in the field of medical imaging, with a wide array of potential applications in clinical diagnosis spanning various disorders and anatomical regions. The process of aligning two images into a common coordinate system enables the detection of subtle changes over time, providing invaluable insights for medical professionals [6]. The core objective of image registration algorithms is to compute transformations that establish correspondence between the two images, allowing for accurate comparison and analysis. This paper has undertaken a comprehensive review of the existing literature on image registration methods, serving as a valuable resource for researchers seeking to implement alternative techniques tailored to specific clinical applications. By consolidating and evaluating a diverse range of methods, this review aims to empower researchers in making informed decisions about the most appropriate image registration approach for their particular medical imaging studies. Ultimately, this compilation of knowledge is expected to contribute significantly to the advancement of medical imaging and enhance the accuracy and effectiveness of clinical diagnoses.

## **7. A review of cardiac image registration methods**

In conclusion, this paper provides a comprehensive review of the current state of cardiac image registration methods. The integration of information from various cardiac imaging modalities, including MRI, CT, PET, SPECT, and ultrasound, holds increasing importance in the medical field for both physiological understanding and diagnostic purposes [7]. However, the registration of cardiac images presents a more intricate challenge compared to brain image registration. This is primarily due to the dynamic and nonrigid nature of the heart, which resides within a body in motion. Furthermore, the heart offers far fewer distinct anatomical landmarks for accurate registration in comparison to the brain. In clinical practice, physicians often rely on mental integration of information from different modalities. Nevertheless, the adoption of automatic registration facilitated by computer programs holds the potential to deliver superior accuracy, repeatability, and time-saving benefits. As such, further advancements in cardiac image registration methods have the potential to significantly enhance the precision and efficiency of diagnostic processes in clinical settings.

## **8. A Survey of Image Registration techniques**

In conclusion, this paper highlights the fundamental role of image registration in various fields of image processing, including target recognition, satellite image analysis, autonomous navigation, and medical diagnostics. Image

registration is essential for aligning images taken at different times, from different sensors, or from different viewpoints. The paper categorizes registration techniques based on three major types of variations in the images. The first type pertains to misalignments caused by differences in image acquisition, which can be corrected by finding an appropriate spatial transformation [8]. The choice of transformation class depends on the specific characteristics of these variations, influencing the overall registration approach. The second type of variations encompasses factors like lighting and atmospheric conditions, which affect intensity values and may introduce spatial distortions. These variations are challenging to model but are important considerations in registration. The third type of variations involves changes of interest within the images, such as object movements or growth. Unlike the first type, these variations should not be entirely removed by registration, as they are pertinent to the analysis. Understanding the characteristics of each type of variation guides the selection of feature space, similarity measure, search space, and search strategy in the registration process. All registration techniques can be viewed as combinations of these choices. This framework provides a valuable perspective for comprehending the strengths and relationships between a wide range of existing registration techniques. It also aids in selecting the most appropriate technique for specific image processing tasks. Overall, this paper contributes to a deeper understanding of image registration and its applications across diverse domains.

## **9. A critical review of image registration methods**

In conclusion, this article delves into the crucial process of image registration, which involves the precise alignment of two or more images of the same area by identifying and aligning common features or control points. The process encompasses four key steps: feature detection, feature matching, transformation function estimation, and image resampling. Image registration finds wide applications in fields like photogrammetry, remote sensing, computer vision, pattern recognition, and medical image registration [9]. This article places particular emphasis on feature point detection and matching, which are pivotal stages in the image registration pipeline. By providing an in-depth review of various techniques, the article aims to equip readers with a comprehensive understanding of the advancements in this domain. Additionally, it serves as a valuable reference for those engaged in related research endeavors. Overall, this article contributes to the broader body of knowledge in image registration, offering insights into the technical intricacies and advancements in this critical area of image processing and analysis. It is anticipated to be a valuable resource for researchers and practitioners alike, facilitating further progress in the field.

## **10. An Overview of Medical Image Registration Methods**

In conclusion, this paper provides a comprehensive overview



of medical image registration methods, offering a systematic classification based on nine key criteria. The primary division within this classification system distinguishes between extrinsic and intrinsic methods. The statistical analysis of this classification reveals discernible trends in the evolution of registration techniques, which are discussed within the paper. Notably, the current landscape of intrinsic methods predominantly falls into two categories: those relying on segmented points or surfaces, and those striving to leverage the complete information content present in the images under consideration [10]. This paper contributes to the understanding of medical image registration by shedding light on the diverse approaches and strategies employed in this critical field. As medical imaging continues to advance, the insights provided here serve as a valuable resource for researchers, practitioners, and those seeking to stay abreast of developments in medical image registration. The classification framework presented here offers a structured perspective on the techniques used to align medical images, ultimately facilitating the pursuit of improved diagnostic and analytical outcomes in healthcare and medical research.

## 11. Medical Image Registration: A Review

In conclusion, this paper offers a comprehensive review of automated image registration methodologies in the medical field. The primary objective is to serve as an introductory resource, providing insight into the breadth of work that has been undertaken in this domain. Additionally, the paper aims to be a valuable reference for individuals seeking suitable registration methods for specific applications in the medical field. The reviewed registration methodologies are categorized into two main classes: intensity-based and feature-based methods [11]. The paper systematically introduces and describes the core steps of these methodologies, including the common geometric transformations employed, the selection of similarity measures, and techniques for assessing accuracy. By providing this structured overview, the paper equips researchers and practitioners with a foundational understanding of the diverse approaches to automated image registration in the medical field. It serves as a valuable starting point for those embarking on registration-related projects and offers a reliable reference for selecting appropriate methodologies for specific medical imaging applications. Overall, this paper contributes to the advancement of medical image registration by consolidating and presenting key methodologies and their associated components.

## 12. A review of recent range image registration methods with accuracy evaluation

In conclusion, this paper addresses the critical issue of three-dimensional reconstruction of real objects in computer vision. Many acquisition systems are inherently limited, often resulting in partial views of the object and leaving blind areas and occlusions. However, in numerous applications,

achieving a full reconstruction is imperative [12]. To tackle this, various techniques have been proposed to fuse 3D surfaces by discerning the motion between different views. The paper focuses on two key challenges in this context. The first pertains to obtaining a rough registration when motion information is not readily available. The second centers on refining this registration from an initial approximation. The paper provides a comprehensive survey of the most prevalent techniques addressing these challenges. To further enhance understanding and facilitate practical implementation, the authors have gone a step further by programming a selection of these techniques and presenting experimental results. These experiments evaluate the performance of these methods in the presence of noise and outliers, offering valuable insights for interested readers. Additionally, the authors have generously provided a Matlab toolbox available on their webpage, facilitating further research and implementation in this area. In sum, this paper not only provides a comprehensive survey of techniques for 3D reconstruction but also offers practical insights through experimentation. It serves as a valuable resource for researchers and practitioners in the field of computer vision, providing both a theoretical foundation and practical tools for advancing the state of the art in three-dimensional object reconstruction.

## 13. A Survey of medical Image Registration

In conclusion, this paper provides a comprehensive survey of recent publications (from 1993 onwards) focusing on medical image registration techniques. The survey employs a classification model based on nine key criteria, with a central distinction between extrinsic and intrinsic methods. The statistical analysis of this classification reveals discernible trends in the evolution of registration techniques, which are discussed within the paper. Notably, the current landscape of intrinsic methods primarily centers around two approaches: those based on segmented points or surfaces, and those striving to leverage the complete information content of the involved images [13]. This survey serves as a valuable resource for researchers and practitioners in the field of medical imaging, offering insights into the diverse approaches and strategies used in image registration. By categorizing and analyzing recent publications, the paper provides a structured overview of the advancements in medical image registration, ultimately contributing to the enhancement of diagnostic and analytical capabilities in healthcare and medical research.

## 14. Improving resolution by Image Registration

In conclusion, this paper highlights the potential for enhancing image resolution by leveraging accurate knowledge of relative displacements in image sequences, coupled with a understanding of the imaging process. The approach presented draws parallels to the back-projection technique commonly employed in tomography [14]. The paper

demonstrates successful applications of this approach in improving the resolution of both gray-level and color images, where previously unknown image displacements are computed from the image sequence. Overall, this research presents a promising avenue for advancing image resolution, with practical implications for various fields reliant on high-quality imaging. By harnessing the power of accurate displacement information and a nuanced understanding of the imaging process, this approach holds significant potential for improving the clarity and detail of images in a range of applications.

## 15. Deep learning in medical Image Registration: A Review

In conclusion, this paper offers a thorough review of deep learning (DL)-based medical image registration methods, providing a comprehensive overview of the latest developments and applications in the medical field. The methods were systematically classified into seven categories based on their techniques, functions, and popularity [15]. Each category was extensively reviewed, highlighting key contributions and identifying specific challenges associated with each approach. A concise assessment followed each detailed review, summarizing the achievements and outlining the potential for future advancements. Furthermore, the paper conducted a comprehensive comparison of DL-based methods for lung and brain registration using benchmark datasets, providing valuable insights into their performance. Additionally, the statistical analysis of cited works shed light on the popularity and emerging trends in DL-based medical image registration. Overall, this paper serves as a valuable resource for researchers and practitioners in the field of medical image registration, offering a comprehensive understanding of the state-of-the-art DL-based approaches and their potential for future developments. The review not only consolidates the latest advancements but also identifies avenues for further research and innovation in this critical area of medical imaging.

## 4 Methodology

### A. Existing

Brain Tumour segmentation methods can be divided as three parts. Manual methods, Semi-automatic methods and Absolute automatic methods. We can determine it according to the level of user interaction required.

#### (i) *Manual methods for Segmentation*

There is a need for medical professionals who can utilize the various information obtained from MRI images and the anatomical and physiological knowledge gained through training and experience. This His procedure requires an expert to go through several His image sections piece

by piece, analyze the brain tumor, and manually cut out the tumor area carefully. This is a time-consuming task, as manual segmentation also relies on the clinician and segmentation results have high variability within and between conversations. However, this is often used to perform his semi-automatic and fully automatic techniques results.

#### (ii) *Semi-automatic methods for Segmentation*

User response is required for three main purposes. Initialization, intervention, or feedback response and evaluation. initialization is primarily performed by defining a region of interest (ROI) that limits the estimated tumor region for processing by the automatic algorithm. The parameters of the preprocessing technique can also be adjusted to suit the input image. In addition to initialization, receiving feedback allows automatic algorithms to be tailored to the desired results throughout the procedure. Appropriate adjustments are also made during this process. Again, users can estimate the results and modify or repeat the process if they are not satisfied. The results are then combined to obtain the final tumor volume. Recent semi-automatic methods have been applied to new classification approaches . In this technique, the segmentation problem is transformed into his classification problem, and brain tumors are segmented by training the to classify only within the same brain. Training machine learning classification techniques for brain tumor segmentation typically requires a large number of brain MRI scans ( answers verified) from a variety of cases. The result primarily deals with correction of intensity distortions and other disturbances. In this his approach, the user initializes the procedure by classifying a subset of voxels associated with each tissue type from a single case. For a subset of these voxels, the algorithm extracts the intensity values and spatial coordinates as features and applies a support vector machine (SVM) that is used to assign all voxels in the same image to the corresponding tissue type. train. In addition to being more time-consuming than manual methods, semi-automated approaches to brain tumor segmentation can maintain efficient results but are still prone to intra- and inter-rater user variability. Therefore, recent research on brain tumor segmentation has mainly focused on fully automated methods.

#### (iii) *Absolute Automatic methods for Segmentation*

This approach requires no user interaction. Most importantly, artificial intelligence and preparatory knowledge can be integrated to solve segmentation problems. Automated segmentation of gliomas, a type of tumor that occurs in the brain and spinal cord, is a very complex and important problem.

#### (iv) *Machine Learning methods*

Supervised learning algorithms like Support Vector Machines (SVM), Random Forests, or Decision Trees are

trained on labeled datasets for microbial classification. Effective for classifying different types of microbes based on learned patterns, requiring annotated datasets for training.

(v) *Deep Learning methods*

Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or more advanced architectures like U-Net are used for end-to-end learning from images. Deep learning excels in feature learning and can automatically extract hierarchical representations, making it powerful for complex microbial detection tasks.

**B. Proposed:**

**1. Chi-Squared Shift, Cross Correlation-Based Shift, and Optical Flow-Based Shift image registration algorithms**

1. **Dataset Acquisition:** Begin by collecting a diverse dataset of image sequences capturing microbes in various environments and conditions. This dataset will serve as the foundation for testing and evaluating the effectiveness of the chosen image registration algorithms.

2. **Image Preprocessing:** Apply preprocessing techniques to the acquired images. This includes tasks such as noise reduction, contrast enhancement, and any necessary adjustments to ensure optimal image quality for subsequent registration processes.

3. **Algorithm Implementation:** Implement the three selected image registration algorithms: Chi-Squared Shift, Cross Correlation-Based Shift, and Optical Flow-Based Shift. Each algorithm will be coded and configured to operate in a parallel computing environment.

4. **Parallel Processing Setup:** Establish a parallel processing framework that enables the simultaneous execution of all three algorithms. This will involve setting up the necessary computing resources and optimizing them for efficient parallel execution.

5. **Concurrent Algorithm Execution:**

Integrate the preprocessed images with the parallel processing framework. Run the selected registration algorithms concurrently on the dataset, allowing them to independently align the images.

6. **Result Multiplotting:**

After the registration process, generate visual outputs by multiplotting the registered images obtained from each algorithm. This step will facilitate direct visual comparison of the results produced by the different algorithms.

7. **Quantitative Evaluation:**

Quantitatively assess the accuracy of the registered images using established metrics. Measure parameters such as

alignment error, overlap, and similarity indices to objectively compare the performance of the algorithms.

8. **Qualitative Evaluation:**

Conduct a qualitative evaluation by visually inspecting the registered images. This subjective assessment will provide insights into the visual quality and alignment precision achieved by each algorithm.

9. **Efficiency Analysis:**

Measure the computational time and resource usage for each algorithm during the registration process. Compare factors such as execution time, memory utilization, and CPU/GPU load to determine the most efficient approach.

10. **Data Analysis and Comparative Study:** Analyze the quantitative and qualitative results obtained from the three algorithms. Perform a comprehensive comparative study to identify the algorithm that demonstrates the highest accuracy and efficiency in microbe detection.

11. **Documentation and Reporting:**

Document the entire implementation process, including codebase, configurations, and experimental setups. Compile the results, analyses, and findings into a comprehensive report.

12. **Conclusion and Recommendations:** Summarize the outcomes of the project, highlighting the strengths and weaknesses of each algorithm. Based on the results, provide recommendations for the most effective approach to microbe detection through image registration. For chi-squared based shift refer [Figure 5](#), for cross-correlation based shift refer [Figure 6](#) and for optical flow based shift refer [Figure 7](#).

**2. Enhanced Approach with CNN and Transfer Learning Models**

**(i) Basic CNN Model:** In our proposed technique for microbe detection, we implemented a Convolutional Neural Network (CNN) to exploit the spatial features inherent in microscopic images. The TensorFlow library was employed for this purpose. The initial steps involved the utilization of an image data generator to perform data augmentation on the training set. We rescaled the pixel values to a range between 0 and 1 using the rescale parameter, applied horizontal flipping for additional training set diversity, and introduced a shear transformation with a range of 0.2 to enhance the model's generalization. Similarly, the test set underwent rescaling to maintain consistency between the datasets. Subsequently, we prepared the training and test datasets using the `flow_from_directory` method, specifying the directory paths, target image size of 120x120 pixels, class mode as categorical and a batch size of 8. Class indices were then extracted for reference in both sets.

Moving to the CNN architecture, we constructed a sequential



model using the Keras API. The first layer consisted of 20 convolutional filters of size (3,3), employing the Rectified Linear Unit (ReLU) activation function to introduce non-linearity. This layer was followed by max-pooling with a pool size of (2,2) to downsample the spatial dimensions. The resulting feature maps were flattened to a one-dimensional array, and two densely connected layers were added subsequently. The first dense layer comprised 45 neurons with ReLU activation, serving as a feature learning stage, while the final dense layer consisted of 4 neurons corresponding to the output classes, activated by the softmax function for probability distribution. For model training, the Adam optimizer was employed with categorical crossentropy as the loss function, and accuracy was chosen as the metric for model evaluation. The training process was executed over 10 epochs, during which the model learned to discern patterns in the augmented training set while being validated on the test set to ensure generalization. Finally, the trained model was saved as 'TARPCNN.h5' for potential deployment and further analysis. This comprehensive approach underscores the significance of our proposed technique, blending data augmentation strategies with deep learning through CNNs to achieve accurate microbe detection in microscopic images. The experimental setup and detailed architecture contribute to the transparency and reproducibility of our methodology in the pursuit of advancing microbial analysis in diverse applications. For the flow chart of CNN model refer [Figure 2](#). For CNN model Initial Phase refer [Figure 8](#). For CNN model training phase refer [Figure 9](#). For CNN model Testing phase and Output refer [Figure 10](#). For CNN model Testing phase and output that identifies the type of cancer refer [Figure 11](#).

## (ii) CNN Upgradation through Layer Increment:

In the refinement of our Convolutional Neural Network (CNN) architecture for microbe detection, we introduced a modified model denoted as modelone. This updated structure reflects a deliberate evolution from the initial model, incorporating adjustments aimed at enhancing the network's capacity for feature extraction and discrimination. The first layer of modelone consists of 12 convolutional filters of size (3,3) with Rectified Linear Unit (ReLU) activation, operating on input images of dimensions 120x120 pixels and three color channels. This is followed by a max-pooling layer with a pool size of (2,2), effectively downsampling the spatial dimensions of the feature maps. The subsequent layers follow a similar pattern, with two additional convolutional layers, each augmented by max-pooling operations, incrementing the number of filters to 24 and 36, respectively. This hierarchical arrangement is designed to capture increasingly complex and abstract features as the spatial resolution diminishes. The Flatten() layer serves to transform the multidimensional output from the convolutional layers into a one-dimensional array, facilitating the transition to densely connected layers. In

modelone, we introduced three densely connected layers, each with an increased number of neurons compared to the previous model. The first dense layer comprises 62 neurons, followed by layers with 32 and 16 neurons, all activated by the ReLU function. This strategic augmentation in the number of neurons aims to enhance the model's capacity to learn intricate patterns and representations from the feature space. The final dense layer, with four neurons and a softmax activation function, continues to produce output probabilities across the four classes, maintaining consistency with the microbe detection task. As in the previous model, the model parameters were optimized using the Adam optimizer, categorical crossentropy served as the loss function, and accuracy was monitored during training. The training phase of modelone involved feeding the augmented training set through the network in batches of eight, validating the model's performance on the test set, and iterating over ten epochs. This training regimen ensures the convergence of the model and its ability to generalize well to unseen data. In summary, the adjustments made in "modelone" aim to refine the CNN architecture by introducing deeper layers, increasing the number of filters, and enhancing the complexity of the densely connected layers. These modifications are undertaken with the objective of improving the model's ability to discern nuanced features in microbe images, ultimately contributing to the efficacy of our proposed technique for microbe detection. For CNN "modelone" Layers refer [Figure 12](#). For CNN "modelone" training phase refer [Figure 13](#). For CNN "modelone" output refer [Figure 14](#).

## (iii) CNN Upgradation using Batch Normalization and Early Stopping: Model Three

In the continuous refinement of our Convolutional Neural Network (CNN) architecture for microbe detection, we present an evolved model denoted as "modelthree". This iteration introduces additional layers and techniques to further enhance the model's robustness, generalization capabilities, and resistance to overfitting. The initial layers of modelthree maintain the convolutional and max-pooling structure from the preceding models, with the key addition of Batch Normalization and Dropout layers after each convolutional layer. Batch Normalization is integrated to normalize and stabilize the activations, addressing issues related to internal covariate shift and promoting a more stable training process. Concurrently, Dropout layers with a rate of 0.2 are introduced to mitigate overfitting by randomly deactivating a fraction of neurons during training, encouraging the network to learn more resilient and generalized features. Similar to the previous models, the convolutional layers are progressively deepened, incrementing the number of filters to 24 and 36, respectively. This hierarchical arrangement ensures the extraction of increasingly complex features while the max-pooling layers effectively downsample the spatial dimensions. The subsequent Flattening layer transforms the multidimensional output from the convolutional layers into a



one-dimensional array, facilitating the transition to densely connected layers. Here, the first densely connected layer with 62 neurons is supplemented by Batch Normalization and a Dropout rate of 0.1, further enhancing the model's capacity for feature learning and generalization. The subsequent densely connected layers, with 32 and 16 neurons, maintain the ReLU activation function, contributing to the network's ability to capture intricate patterns. Finally, the output layer retains the softmax activation function with four neurons, aligning with the microbe detection task's multi-class classification nature. As before, the model parameters are optimized using the Adam optimizer, categorical crossentropy serves as the loss function, and accuracy is monitored during training. To manage the training process effectively, an Early Stopping callback is introduced, monitoring the accuracy metric and halting the training process if no improvement is observed over a specified patience period of eight epochs. This proactive approach prevents overfitting and streamlines model convergence. The training process was executed over 20 epochs, during which the model learned to discern patterns in the augmented training set while being validated on the test set to ensure generalization. In summary, modelthree represents a significant evolution in our CNN architecture, incorporating Batch Normalization and Dropout layers to address overfitting concerns and enhance stability during training. The introduction of these techniques, along with the progressively deepened convolutional layers, contributes to a more resilient and accurate microbe detection model, underscoring our commitment to continuous improvement in the pursuit of advanced microbial analysis. For CNN "modelthree" Layers refer [Figure 15](#). For CNN "modelthree" training phase refer [Figure 16](#). For CNN "modelthree" output refer [Figure 17](#).

#### (iv) Transfer Learning - VGG16:

In our proposed technique for microbe detection, transfer learning emerges as a key strategy to enhance the efficiency and effectiveness of our convolutional neural network (CNN). Transfer learning leverages the knowledge acquired by a pre-trained model, such as VGG16, on a large-scale dataset like ImageNet, and adapts it to a specific task with a more limited dataset. Compared to training a CNN from scratch, transfer learning allows for quicker convergence and often results in improved generalization on a smaller dataset, making it a valuable approach for our microbe detection challenge. Contrasting this with the conventional CNN models described earlier, where we meticulously designed and trained each architecture from the ground up, the introduction of VGG16 in this code signifies a departure from this approach. Instead, we incorporate a pre-trained model with a well-established architecture, benefiting from its ability to capture hierarchical features from diverse images. By freezing the majority of the VGG16 layers, we retain its learned representations and customize the top layers

to suit the microbe detection task. This adaptation aligns with our commitment to efficiency and resource-conscious methodologies in our proposed technique. VGG16, a well-known architecture, is employed as the base model in this scenario, with its top layers removed. These layers are replaced with a new set of densely connected layers tailored to the microbe detection task. For flow chart refer [Figure 3](#).

The provided code implements transfer learning using the VGG16 architecture for microbe detection. Firstly, the paths to the training and testing datasets are specified. ImageDataGenerators are then employed to preprocess the data by rescaling pixel values, applying shear and zoom transformations, and horizontal flipping for increased training set diversity. The VGG16 model, pre-trained on the ImageNet dataset, is instantiated with its top layers excluded. The architecture is frozen to prevent further training on these layers. The model is modified by adding a Flatten layer and a new Dense layer with softmax activation for the specific microbe detection task. The resulting model, referred to as modelfour, is compiled using categorical crossentropy loss and the Adam optimizer. Training is performed using the fit\_generator method on the generators created from the training and testing datasets. This method allows for data augmentation during training. Alternatively, the model can be trained using the fit method, as shown in the latter part of the code. This process involves fine-tuning the new top layers of the VGG16 model on the microbe detection dataset. The model initiates the training of the modelfour neural network. Utilizing the trainvalue dataset in batches of 8 samples per iteration, the model learns from the data for ten epoch. The validation\_data parameter evaluates the model's performance on the separate testvalue dataset, providing insights into its ability to generalize to new data and potential overfitting concerns. Adjusting the number of epochs influences the duration and depth of the training process, impacting the model's learning and adaptation to the microbe detection task. In summary, the code demonstrates the utilization of transfer learning with the VGG16 architecture for microbe detection, showcasing the advantages of leveraging pre-trained models for improved efficiency and performance compared to training models from scratch. The introduction of transfer learning represents a significant departure from the previous CNN models, emphasizing a more sophisticated approach to feature extraction and model initialization.

**(v) Transfer Learning - ResNet-50:** In our continuous pursuit of refining microbe detection models, we have incorporated the ResNet50 architecture into our proposed technique for transfer learning. In this approach, we leverage the ResNet50 model, which has been pre-trained on extensive datasets, allowing it to capture intricate hierarchical features. Upon instantiation, the top layers of ResNet50 are excluded, and its convolutional layers are marked

as non-trainable, preserving the learned features. This strategic choice aligns with our commitment to efficiency and resource-conscious methodologies, allowing the model to capitalize on pre-existing knowledge. Two loops iterate through the layers of the ResNet50 model, ensuring that they remain non-trainable. By doing so, we retain the robust features acquired during pre-training on diverse image data. Subsequently, a Flatten layer is introduced to transform the multidimensional output into a one-dimensional array, facilitating the incorporation of a new Dense layer designed for the microbe detection task. The resulting model, denoted as `res_model`, is compiled with categorical crossentropy loss and the Adam optimizer, aligning with the standards set in our proposed technique. In the training phase, the `res_model` is fit to the training data (`trainvalue`) for ten epochs, with each epoch representing a complete iteration through the entire training dataset. The validation is performed on the test data (`testvalue`). The use of `steps_per_epoch` and `validation_steps` parameters ensures that the entire training and validation datasets are processed in each epoch. This iterative process allows the model to adapt and refine its parameters over multiple passes through the data, contributing to improved microbe detection accuracy. Comparing the ResNet50 model with our previous four models (`model`, `modelone`, `modelthree`, and VGG16), the adoption of ResNet50 marks a departure from the conventional architectures previously employed. ResNet50's unique architecture, featuring residual connections, addresses vanishing gradient issues, potentially allowing for more effective training and feature extraction. This contrasts with the sequential architectures of the earlier models, showcasing our adaptability and willingness to explore advanced model architectures in the pursuit of optimal microbe detection performance.

#### **(vi) Transfer Learning - Inception v3:**

In our ongoing effort to advance microbe detection, we have incorporated the InceptionV3 architecture into our proposed technique, embracing the advantages of transfer learning. The base model is instantiated with pre-trained weights from ImageNet, and its top layers are excluded, preserving the convolutional layers' learned features. In line with our proposed technique, we set the layers of the InceptionV3 model as non-trainable to harness its pre-existing knowledge effectively. Two loops iterate through the layers of the InceptionV3 model, securing their non-trainable status. This meticulous preservation of pre-trained features ensures the retention of rich hierarchical representations from diverse image data. The Global Average Pooling layer is introduced to compress the spatial dimensions of the output, preparing it for the subsequent addition of a Dense layer tailored to the microbe detection task. The resulting model, denoted as `model_i`, is compiled with categorical crossentropy loss and the Adam optimizer, aligning with our proposed technique's standards. In the training phase, the `model_i` is fit to the training data (`trainvalue`) for ten epochs. Each

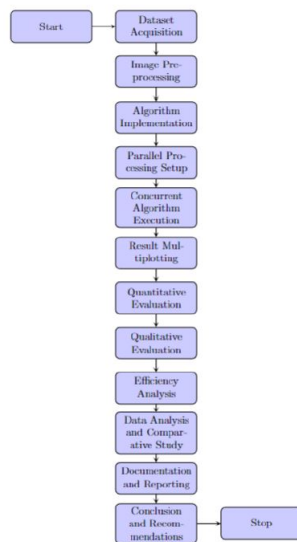
epoch represents a complete iteration through the entire training dataset. Validation is performed on the test data (`testvalue`). The use of `steps_per_epoch` and `validation_steps` parameters ensures the comprehensive processing of the training and validation datasets in each epoch, contributing to the model's adaptability and enhanced microbe detection accuracy. Comparing the InceptionV3 model with our previous five models (`model`, `modelone`, `modelthree`, VGG16, and ResNet50), the adoption of InceptionV3 introduces a distinctive architecture characterized by inception modules, allowing for the extraction of features at multiple scales. This contrasts with the sequential architectures of the earlier models, showcasing our adaptability and commitment to exploring diverse model architectures within our proposed technique. The InceptionV3 model's unique design may contribute to improved performance in microbe detection tasks, and its incorporation reflects our ongoing exploration of state-of-the-art techniques for enhanced accuracy and robustness.

#### **(vii) Transfer Learning - Xception:**

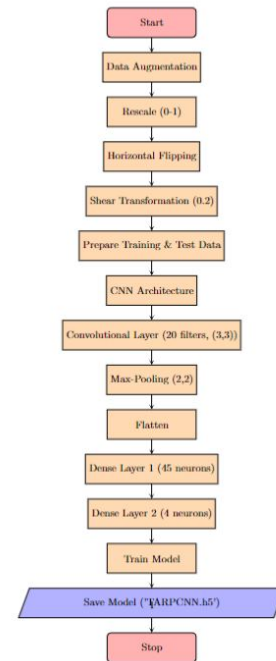
In our pursuit of advancing microbe detection models within our proposed technique, we have integrated the Xception architecture. This represents a strategic decision to leverage the advantages of transfer learning, specifically with the Xception model, renowned for its depthwise separable convolutions. The paths to the training and testing datasets are specified, and ImageDataGenerators are employed for data preprocessing, adhering to the standards set in our proposed technique. The Xception model is instantiated with pre-trained weights from ImageNet, and its top layers are excluded to retain the valuable features learned from a diverse range of images. Notably, the input shape is set to (299, 299, 3), reflecting the specific requirements of Xception. To align with the model's architecture, which utilizes depthwise separable convolutions, we adjust the image size to 299x299 pixels. This demonstrates our commitment to model-specific considerations within our proposed technique. Two loops iterate through the layers of the Xception model, marking them as non-trainable. This meticulous preservation of pre-trained features is essential for effective knowledge transfer. The introduction of a Global Average Pooling layer compresses the spatial dimensions of the output, and a subsequent Dense layer with 512 neurons and ReLU activation enhances feature representation. The final Dense layer, tailored for the microbe detection task, employs softmax activation to output probabilities for each class. The resulting model, referred to as `model_q`, is summarized to provide insights into its architecture. It is then compiled using categorical crossentropy loss and the Adam optimizer, adhering to the standards set in our proposed technique. In the training phase, the `model_q` is fit to the training data (`trainvalueone`) for ten epochs, aligning with the specific requirements of Xception. Each epoch represents a complete iteration through the entire training dataset, and

validation is performed on the test data (testvalueone). The use of `steps_per_epoch` and `validation_steps` parameters ensures comprehensive processing of the training and validation datasets in each epoch, contributing to the model's adaptability and enhanced microbe detection accuracy. Comparing the Xception model with our previous six models (model, modelone, modelthree, VGG16, ResNet50, and Inception), the introduction of Xception represents a departure from conventional architectures. The adjustment of image sizes to 299x299 pixels is a model-specific adaptation, showcasing our meticulous consideration of architecture requirements within our proposed technique. This reflects our commitment to innovation and adaptability in the pursuit of optimal microbe detection performance.

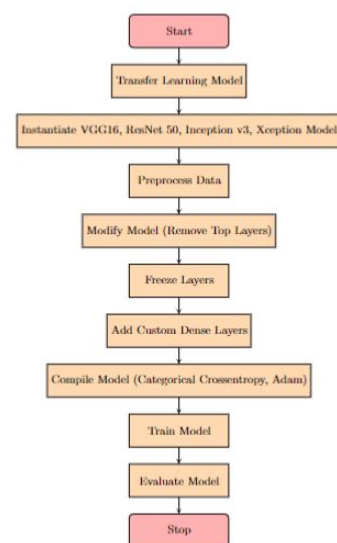
### C. Flow Chart



**Figure 1: Image Registration**



**Figure 2: CNN Model**

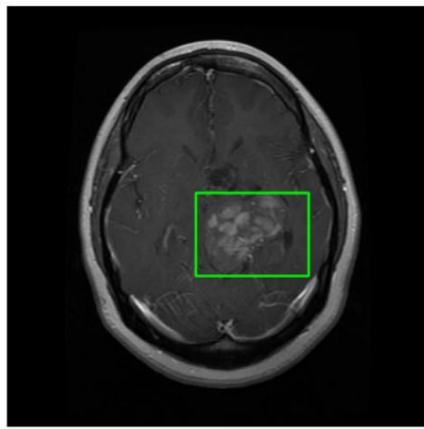


**Figure 3: Transfer Learning**

## 5 Results and Discussion

### (i) Image Registration:

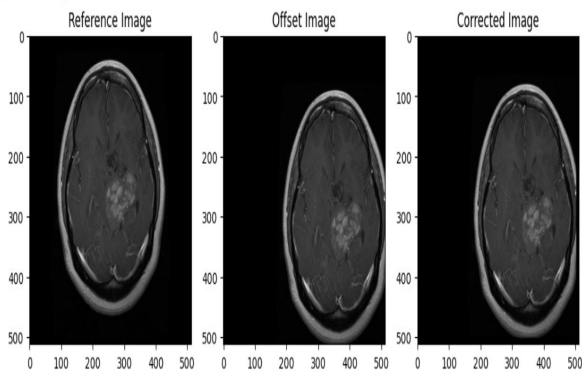
After using the image registration technique along with the CNN model, we are able to identify tumour in the scanned brain MRI images. Figure 4 depicts the brain tumour detection.



Offset (x, y): 229 227

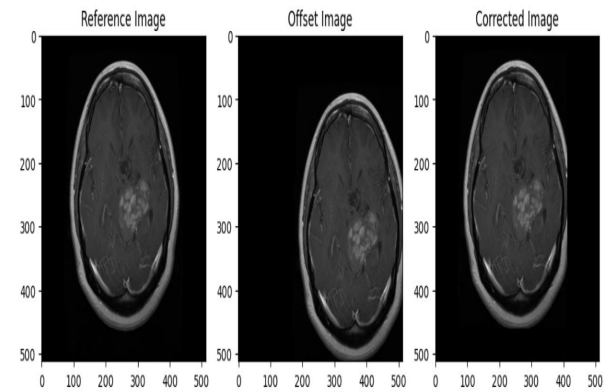
**Figure 4: Brain Tumour Detection**

Optimal Shift in X-axis: -10  
Optimal Shift in Y-axis: -10  
Optimal Chi-squared Statistic: 39951.65768709922



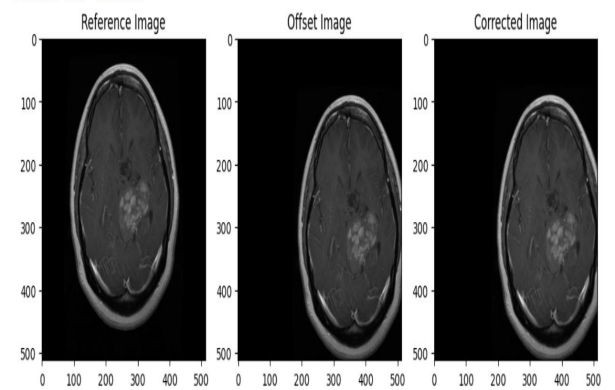
**Figure 5: Chi-Squared based Shift**

Shift in x-axis: -181  
Shift in y-axis: -51



**Figure 6: Cross-Correlation based shift**

Shift in X-axis: 8.420584  
Shift in Y-axis: -0.29750837



**Figure 7: Optical flow based shift**



## (ii) CNN Model Interface:

In our proposed model, we used flask to integrate front-end and model weights. We created the interface so that user can upload the MRI images and get the desired cancer classification results. Now refer [Figure 18](#) for the website interface. Refer figures: [Figure 19](#), [Figure 20](#), [Figure 21](#) and [Figure 22](#) for output of cancer classification.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_data = ImageDataGenerator(rescale=1./255, horizontal_flip=True, shear_range=0)
test_data = ImageDataGenerator(rescale=1./255)

trainset = train_data.flow_from_directory('content/training_data', target_size=(128, 128), class_mode='categorical', batch_size=32)
testset = test_data.flow_from_directory('content/test_data', target_size=(128, 128), class_mode='categorical', batch_size=32)

# Load 300 images belonging to 4 classes.
# Load 300 images belonging to 4 classes.

trainset.class_indices
{'Brain Tumor': 0, 'Breast Cancer': 1, 'Healthy': 2, 'Lung Cancer': 3}

testset.class_indices
{'Brain Tumor': 0, 'Breast Cancer': 1, 'Healthy': 2, 'Lung Cancer': 3}

from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.models import Sequential
```

Figure 8: CNN model initial phase

```
model = Sequential()
model.add(Convolution2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(1000, activation='relu'))
model.add(Dense(10, activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

model.fit(trainset.batch_size(32), validation_data=testset, epochs=10)

Epoch 1/10
128/128 [=====] - loss: 0.1718 - accuracy: 0.8020 - val_loss: 0.8062 - val_accuracy: 0.8993
Epoch 2/10
128/128 [=====] - loss: 0.1296/step - loss: 7.5976e-04 - accuracy: 1.0000 - val_loss: 0.8482 - val_accuracy: 0.8993
Epoch 3/10
128/128 [=====] - loss: 0.1126/step - loss: 7.3427e-04 - accuracy: 1.0000 - val_loss: 0.8623 - val_accuracy: 0.8993
Epoch 4/10
128/128 [=====] - loss: 0.1296/step - loss: 1.4523e-04 - accuracy: 1.0000 - val_loss: 0.8497 - val_accuracy: 0.8975
Epoch 5/10
128/128 [=====] - loss: 0.1296/step - loss: 7.5838e-05 - accuracy: 1.0000 - val_loss: 0.8492 - val_accuracy: 0.8993
Epoch 6/10
128/128 [=====] - loss: 0.1296/step - loss: 5.4888e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 7/10
128/128 [=====] - loss: 0.1296/step - loss: 3.7336e-05 - accuracy: 1.0000 - val_loss: 0.8426 - val_accuracy: 0.8993
Epoch 8/10
128/128 [=====] - loss: 0.1296/step - loss: 3.3276e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 9/10
128/128 [=====] - loss: 0.1296/step - loss: 3.3276e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 10/10
128/128 [=====] - loss: 0.1296/step - loss: 1.4753e-05 - accuracy: 1.0000 - val_loss: 0.8526 - val_accuracy: 0.8993
Observe callbacks history at 0x7f96c4d42620
```

Figure 9: CNN model training phase

```
imgfour = image.load_img('content/test_data/Healthy/img_35_m_0_3748.jpg', target_size=(128, 128))
imgfour = image.img_to_array(imgfour)
imgfour = np.expand_dims(imgfour, axis=0)
datafour = np.argmax(model.predict(imgfour))
print(datafour)
finaloutput = ['Brain Tumor', 'Breast Cancer', 'Healthy', 'Lung Cancer']
print(finaloutput[datafour])

1/1 [=====] - loss: 0.1296/step
2
Healthy

[23] model = Sequential()
model.add(Convolution2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(1000, activation='relu'))
model.add(Dense(10, activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

model.fit(trainset.batch_size(32), validation_data=testset, epochs=10)

Epoch 1/10
128/128 [=====] - loss: 0.1718 - accuracy: 0.8020 - val_loss: 0.8062 - val_accuracy: 0.8993
Epoch 2/10
128/128 [=====] - loss: 0.1296/step - loss: 7.5976e-04 - accuracy: 1.0000 - val_loss: 0.8482 - val_accuracy: 0.8993
Epoch 3/10
128/128 [=====] - loss: 0.1126/step - loss: 7.3427e-04 - accuracy: 1.0000 - val_loss: 0.8623 - val_accuracy: 0.8993
Epoch 4/10
128/128 [=====] - loss: 0.1296/step - loss: 1.4523e-04 - accuracy: 1.0000 - val_loss: 0.8497 - val_accuracy: 0.8975
Epoch 5/10
128/128 [=====] - loss: 0.1296/step - loss: 7.5838e-05 - accuracy: 1.0000 - val_loss: 0.8492 - val_accuracy: 0.8993
Epoch 6/10
128/128 [=====] - loss: 0.1296/step - loss: 5.4888e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 7/10
128/128 [=====] - loss: 0.1296/step - loss: 3.7336e-05 - accuracy: 1.0000 - val_loss: 0.8426 - val_accuracy: 0.8993
Epoch 8/10
128/128 [=====] - loss: 0.1296/step - loss: 3.3276e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 9/10
128/128 [=====] - loss: 0.1296/step - loss: 3.3276e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 10/10
128/128 [=====] - loss: 0.1296/step - loss: 1.4753e-05 - accuracy: 1.0000 - val_loss: 0.8526 - val_accuracy: 0.8993
Observe callbacks history at 0x7f96c4d42620
```

Figure 10: CNN model Testing phase

For CNN model Accuracy and Model Loss refer [Figure 23](#). For CNN model Training and Validation Accuracy refer [Figure 24](#). For CNN "modelone" Accuracy and Model Loss refer [Figure 25](#). For Accuracy of the CNN Model refer [Table 1](#).

```
[10] imgfour = image.load_img('content/test_data/Healthy/img_35_m_0_3748.jpg', target_size=(128, 128))
imgfour = image.img_to_array(imgfour)
imgfour = np.expand_dims(imgfour, axis=0)
datafour = np.argmax(model.predict(imgfour))
print(datafour)
finaloutput = ['Brain Tumor', 'Breast Cancer', 'Healthy', 'Lung Cancer']
print(finaloutput[datafour])

1/1 [=====] - loss: 0.1296/step
2
Healthy
```

Figure 11: CNN model Output

```
[23] model = Sequential()
model.add(Convolution2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Convolution2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Convolution2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(1000, activation='relu'))
model.add(Dense(10, activation='softmax'))
model.add(Dense(10, activation='softmax'))
```

Figure 12: CNN modelone Layers

```
[26] modelone.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

[27] modelone.fit(trainset.batch_size(32), validation_data=testset, epochs=10)

Epoch 1/10
128/128 [=====] - loss: 0.1718 - accuracy: 0.8020 - val_loss: 0.8062 - val_accuracy: 0.8993
Epoch 2/10
128/128 [=====] - loss: 0.1296/step - loss: 7.5976e-04 - accuracy: 1.0000 - val_loss: 0.8482 - val_accuracy: 0.8993
Epoch 3/10
128/128 [=====] - loss: 0.1126/step - loss: 7.3427e-04 - accuracy: 1.0000 - val_loss: 0.8623 - val_accuracy: 0.8993
Epoch 4/10
128/128 [=====] - loss: 0.1296/step - loss: 1.4523e-04 - accuracy: 1.0000 - val_loss: 0.8497 - val_accuracy: 0.8975
Epoch 5/10
128/128 [=====] - loss: 0.1296/step - loss: 7.5838e-05 - accuracy: 1.0000 - val_loss: 0.8492 - val_accuracy: 0.8993
Epoch 6/10
128/128 [=====] - loss: 0.1296/step - loss: 5.4888e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 7/10
128/128 [=====] - loss: 0.1296/step - loss: 3.7336e-05 - accuracy: 1.0000 - val_loss: 0.8426 - val_accuracy: 0.8993
Epoch 8/10
128/128 [=====] - loss: 0.1296/step - loss: 3.3276e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 9/10
128/128 [=====] - loss: 0.1296/step - loss: 3.3276e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 10/10
128/128 [=====] - loss: 0.1296/step - loss: 1.4753e-05 - accuracy: 1.0000 - val_loss: 0.8526 - val_accuracy: 0.8993
Observe callbacks history at 0x7f96c4d42620
```

Figure 13: CNN modelone Training phase

```
[28] imgfour = image.load_img('content/test_data/Healthy/img_35_m_0_3748.jpg', target_size=(128, 128))
imgfour = image.img_to_array(imgfour)
imgfour = np.expand_dims(imgfour, axis=0)
datafour = np.argmax(modelone.predict(imgfour))
print(datafour)
finaloutput = ['Brain Tumor', 'Breast Cancer', 'Healthy', 'Lung Cancer']
print(finaloutput[datafour])

1/1 [=====] - loss: 0.1296/step
2
Healthy
```

Figure 14: CNN modelone Output

```
[33] from tensorflow.keras.layers import BatchNormalization, Dropout

[34] modelthree = Sequential()
modelthree.add(Convolution2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)))
modelthree.add(BatchNormalization())
modelthree.add(MaxPooling2D(pool_size=(2, 2)))
modelthree.add(Dropout(0.2))
modelthree.add(Convolution2D(64, (3, 3), activation='relu'))
modelthree.add(BatchNormalization())
modelthree.add(MaxPooling2D(pool_size=(2, 2)))
modelthree.add(Dropout(0.2))
modelthree.add(Flatten())
modelthree.add(Dense(1000, activation='relu'))
modelthree.add(BatchNormalization())
modelthree.add(Dense(10, activation='softmax'))
modelthree.add(Dense(10, activation='softmax'))
modelthree.add(Dense(10, activation='softmax'))
```

Figure 15: CNN modelthree Layers

```
[36] modelthree.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

[37] from tensorflow.keras.callbacks import EarlyStopping

[38] earlystop = EarlyStopping(monitor='accuracy', patience=10)

[39] modelthree.fit(trainset.batch_size(32), validation_data=testset, epochs=10, callbacks=[earlystop])

Epoch 1/10
128/128 [=====] - loss: 0.1718 - accuracy: 0.8020 - val_loss: 0.8062 - val_accuracy: 0.8993
Epoch 2/10
128/128 [=====] - loss: 0.1296/step - loss: 7.5976e-04 - accuracy: 1.0000 - val_loss: 0.8482 - val_accuracy: 0.8993
Epoch 3/10
128/128 [=====] - loss: 0.1126/step - loss: 7.3427e-04 - accuracy: 1.0000 - val_loss: 0.8623 - val_accuracy: 0.8993
Epoch 4/10
128/128 [=====] - loss: 0.1296/step - loss: 1.4523e-04 - accuracy: 1.0000 - val_loss: 0.8497 - val_accuracy: 0.8975
Epoch 5/10
128/128 [=====] - loss: 0.1296/step - loss: 7.5838e-05 - accuracy: 1.0000 - val_loss: 0.8492 - val_accuracy: 0.8993
Epoch 6/10
128/128 [=====] - loss: 0.1296/step - loss: 5.4888e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 7/10
128/128 [=====] - loss: 0.1296/step - loss: 3.7336e-05 - accuracy: 1.0000 - val_loss: 0.8426 - val_accuracy: 0.8993
Epoch 8/10
128/128 [=====] - loss: 0.1296/step - loss: 3.3276e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 9/10
128/128 [=====] - loss: 0.1296/step - loss: 3.3276e-05 - accuracy: 1.0000 - val_loss: 0.8498 - val_accuracy: 0.8993
Epoch 10/10
128/128 [=====] - loss: 0.1296/step - loss: 1.4753e-05 - accuracy: 1.0000 - val_loss: 0.8526 - val_accuracy: 0.8993
Observe callbacks history at 0x7f96c4d42620
```

Figure 16: CNN modelthree Training phase

```
[42] imgfour = image.load_img('content/test_data/Lung Cancer/imgsc4003_3_110m.jpg', target_size=(128, 128))
imgfour = image.img_to_array(imgfour)
imgfour = np.expand_dims(imgfour, axis=0)
datafour = np.argmax(modelthree.predict(imgfour))
print(datafour)
finaloutput = ['Brain Tumor', 'Breast Cancer', 'Healthy', 'Lung Cancer']
print(finaloutput[datafour])

1/1 [=====] - loss: 0.1296/step
2
Lung Cancer
```

Figure 17: CNN modelthree Output

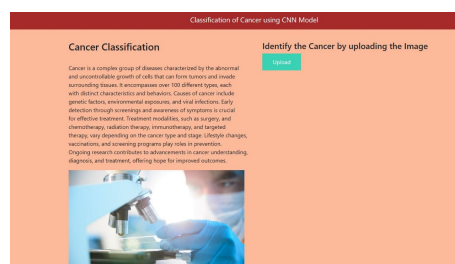
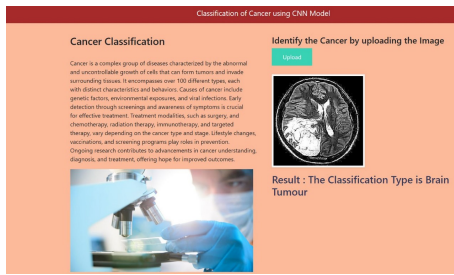
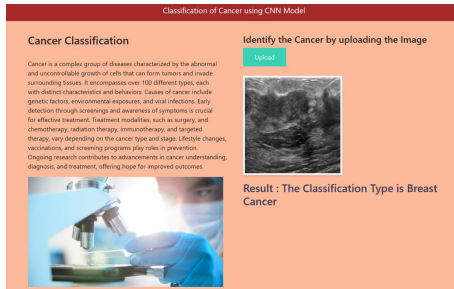


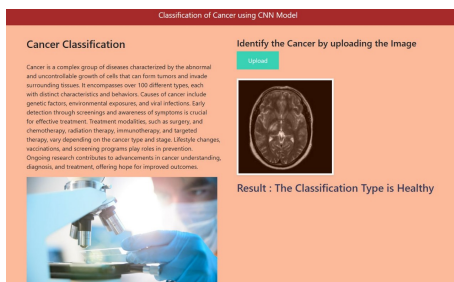
Figure 18: Website Interface



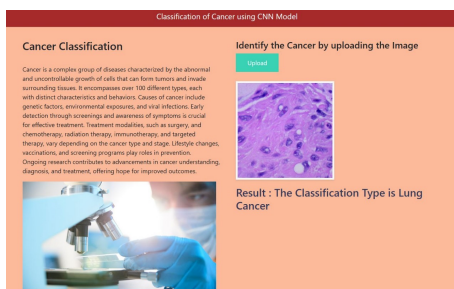
**Figure 19:** Brain Tumour detection output



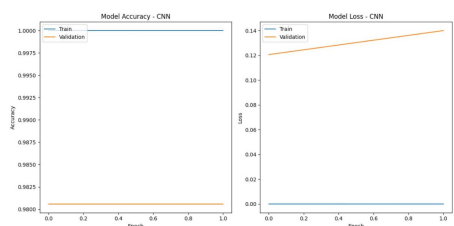
**Figure 20:** Breast cancer detection output



**Figure 21:** Healthy brain output



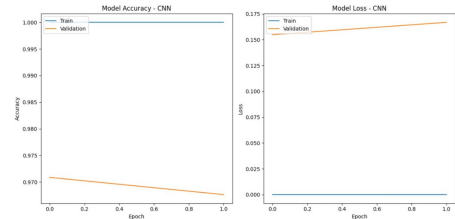
**Figure 22:** Lung Cancer detection output



**Figure 23:** CNN model Accuracy and Model Loss



**Figure 24:** CNN model Training and Validation Accuracy



**Figure 25:** CNN modelone Accuracy and Model Loss

Training Images	Testing Images	Split Ratio	Highest Training Accuracy Recorded	Highest Testing Accuracy Recorded
900	309	3 : 1	100	99.35

**Table 1:** CNN Model Accuracy

## 6 Conclusion and Future Scope

In conclusion, our research introduces a pioneering paradigm that seamlessly integrates image registration techniques with advanced convolutional neural networks (CNNs) employing transfer learning. This unified approach significantly elevates the accuracy and efficacy of medical image analysis, particularly in classifying 'Healthy', 'Brain Tumour', 'Breast Cancer', and 'Lung Cancer'.

At the core of our methodology lies the synergistic combination of image registration techniques and CNNs. Commencing with a foundational CNN architecture, our model undergoes iterative augmentation involving increased convolution and max-pooling layers. The pinnacle of this evolution incorporates batch normalization and early stopping, enhancing the CNN's discernment of complex features crucial for precise microbe classification. The utilization of transfer learning models, including VGG16, ResNet50, Inception, and Xception, further amplifies the CNN's proficiency in discerning medical images.

Simultaneously, our incorporation of image registration algorithms—chi-squared shift, cross-correlation-based shift, optical flow-based shift, and an innovative averaging approach—ensures accurate and consistent alignment of input images. These algorithms play a pivotal role in normalizing

inputs for subsequent CNN analysis, providing a standardized foundation.

The seamless collaboration between image registration and CNNs ensures that registered images serve as optimal inputs for subsequent classification, avoiding redundant processes. This integration leverages the strengths of each methodology, culminating in a synergetic enhancement of microbial detection accuracy.

The encapsulation of our proposed system within a Flask framework facilitates a user-friendly interface, enabling real-time cancer classification upon uploading MRI images. In practical scenarios where manual detection poses significant challenges, our algorithmic approach stands as a valuable solution, contributing to significant advancements in automated microbe detection and diagnosis.

Notably, our model achieved outstanding results with a highest recorded training accuracy of 100 Percentage and testing accuracy of 99.35 Percentage. These impressive outcomes underscore the efficacy of our integrated approach in achieving unparalleled accuracy in microbe detection.

In summary, our research makes substantial contributions to the fields of medical imaging, computer vision, and healthcare technology. By advancing automated microbe detection, our integrated methodology stands as a transformative solution with broad implications for precision diagnostics and patient care.

## 7 References:

- [1] Dhindsa, A., Bhatia, S., Agrawal, S., Sohi, B. S. (2020). *An efficient microbes detection system using microscopic images via morphological and correlation based features. Biomedical and Pharmacology Journal*, 13(3), 1113-1124.
- [2] Bandara, C. D., Schmidt, M., Davoudpour, Y., Stryhanyuk, H., Richnow, H. H., Musat, N. (2021). *High-Resolution Chemical Mapping and Microbial Identification of Rhizosphere using Correlative Microscopy. bioRxiv*, 2021-02.
- [3] van Heel, M., Harauz, G., Orlova, E. V., Schmidt, R., Schatz, M. (1996). *A new generation of the IMAGIC image processing system. Journal of structural biology*, 116(1), 17-24.
- [4] Bliznuks, D., Lihachev, A., Liepins, J., Uteshev, D., Chizhov, Y., Bondarenko, A., Bolochko, K. (2019, June). *Automated microorganisms activity detection on the early growth stage using artificial neural networks. In European Conference on Biomedical Optics (p. 1107560).OpticaPublishingGroup*.
- [5] Zitova, B., Flusser, J. (2003). *Image registration methods: a survey. Image and vision computing*, 21(11), 977-1000.
- [6] Wyawahare, M. V., Patil, P. M., Abhyankar, H. K. (2009). *Image registration techniques: an overview. International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2(3), 11-28.
- [7] Makela, T., Clarysse, P., Sipila, O., Pauna, N., Pham, Q. C., Katila, T., Magnin, I. E. (2002). *A review of cardiac image registration methods. IEEE Transactions on medical imaging*, 21(9), 1011-1021.
- [8] Brown, L. G. (1992). *A survey of image registration techniques. ACM computing surveys (CSUR)*, 24(4), 325-376.
- [9] Xiong, Z., Zhang, Y. (2010). *A critical review of image registration methods. International Journal of Image and Data Fusion*, 1(2), 137-158.
- [10] Maintz, J. A., Viergever, M. A. (1996). *An overview of medical image registration methods. In Symposium of the Belgian hospital physicists association (SBPH/BVZF) (Vol. 12, No. V, pp. 1-22)*.
- [11] Oliveira, F. P., Tavares, J. M. R. (2014). *Medical image registration: a review. Computer methods in biomechanics and biomedical engineering*, 17(2), 73-93.
- [12] Salvi, J., Matabosch, C., Fofi, D., Forest, J. (2007). *A review of recent range image registration methods with accuracy evaluation. Image and Vision computing*, 25(5), 578-596.
- [13] Maintz, J. A., Viergever, M. A. (1998). *A survey of medical image registration. Medical image analysis*, 2(1), 1-36.
- [14] Irani, M., Peleg, S. (1991). *Improving resolution by image registration. CVGIP: Graphical models and image processing*, 53(3), 231-239.
- [15] Fu, Y., Lei, Y., Wang, T., Curran, W. J., Liu, T., Yang, X. (2020). *Deep learning in medical image registration: a review. Physics in Medicine Biology*, 65(20), 20TR01.