PCOS Detection Using MobileNet and Image Processing Techniques

B. Tech Project Report

Submitted to The University College of Engineering and Technology,
Acharya Nagarjuna University in partial fulfilment of
Requirement for the award of Degree

BACHELOR OF TECHNOLOGY
in

COMPUTER SCIENCE AND ENGINEERING

by

SANTHA PALLAVI	(Y20CS3245)
MURIGESHAN MANJUNADHAN	(Y20CS3234)
NAGARAJU DIVYA	(Y20CS3235)
VEMURI VISHNUVARDHAN	(L21CS3276)
PILLI MOHAN KUMAR	(L21CS3273)

Under the Guidance of
Dr. V Balaji, M.Tech, Phd

UNIVERSITY COLLEGE OF ENGINEERING & TECHNOLOGY
ACHARYA NAGARJUNA UNIVERSITY
NAGARJUNA NAGAR -522510, GUNTUR, A.P., INDIA
2020-2024

UNIVERSITY COLLEGE OF ENGINEERING & TECHNOLOGY

ACHARYA NAGARJUNA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"PCOS Detection Using MobileNet and Image Processing Techniques"** is a bona fide record of the project work done by **Nagaraju Divya (Y20CS3235)** under my supervision and guidance, in partial fulfilment of the requirements for the award of Degree in Computer Science & Engineering from University College of Engineering & Technology, Guntur for the academic year 2023-24

…………………………….

Dr Balaji Vicharapu

Project Guide, Dept. of C.S.E

……………………….

Dr Balaji Vicharapu

HOD ,Dept of C.S.E

…………………………

External Examiner

DECLARATION

We hereby declare that the project entitled, **“PCOS Detection Using MobileNet and Image Processing Techniques”** was carried out and written by me under the guidance of **Dr.V.Balaji** , HOD, Department of Computer Science & Engineering, University College of Engineering & Technology, Acharya Nagarjuna University. This work has not been previously formed the basis for the award of any degree or diploma or certificate nor has been submitted elsewhere for the award of any degree or diploma.

Place: ANUCET **Nagaraju Divya (Y20CS3235)**

Date:

ACKNOWLEDGEMENTS

In our opinion writing acknowledgement is a difficult but most beautiful part of the project report ending. It is difficult to express the heartfelt accomplishments and the gratitude for those who really or virtually supported for making the work successful and memorable.

We would like to convey our sincere gratitude and admiration to our Guide **Dr V.Balaji**, University College of Engineering & Technology, Acharya Nagarjuna University (ANU), Guntur, for his/her excellent support and valuable guidance throughout our study. His/her marvelous advice motivated us to work on this, and make it successful. His/her supportive nature and thought-provoking ideas made this work possible.

We are very grateful to **Dr V. Balaji,** HOD, Dept of CSE, University College of Engineering & Technology, Acharya Nagarjuna University, for her keen interest, counselling, constant encouragement, personal affection and valuable suggestions in completion of this research work.

We would like to express our heartfelt thanks to **Prof. P. Siddaiah**, Principal, University College of Engineering & Technology, Acharya Nagarjuna University, Guntur, for guidance, mentorship, and valuable insights that have been instrumental in the successful completion of this work

We would like to express our sincere appreciation and heartfelt gratitude to **Prof E. Sreenivasa Reddy**, Dean Faculty of Engineering, University College of Engineering & Technology, Acharya Nagarjuna University, Guntur, for his unwavering support and encouragement throughout the duration of this project.

We are deeply indebted to **Dr U Satish Kumar**, Project Coordinator, for Conducting reviews and giving constructive criticism throughout the project duration.

We are grateful to the **Authorities** of University College of Engineering & Technology, Acharya Nagarjuna University, enabling us to pursue our work.

We are very thankful to all **Teaching** and **Non-teaching staff** of Department of Computer Science & Engineering, University College of Engineering & Technology, Acharya Nagarjuna University, Guntur, for their encouragement, co-operation and assistance during the work period.

We would like to express our grateful thanks to **our parents** our whole beloved family for their love, unwavering support and encouragement throughout the time of our study.

Finally, we thank one and all, who helped us directly or indirectly for completing this work.

With gratitude
Nagaraju Divya (Y20CS3235)

ABSTRACT

This project focuses on the development and validation of a convolutional neural network (CNN) for the detection of polycystic ovary syndrome (PCOS) using ultrasound images. PCOS is a prevalent endocrine disorder affecting women of reproductive age, characterized by irregular menstrual cycles, polycystic ovaries, and elevated androgen levels. Accurate and timely diagnosis is crucial for effective management and treatment. Our approach utilizes a dataset of ultrasound images categorized into two classes: infected and not infected. We preprocess these images by converting them to grayscale, resizing to 128x128 pixels, and normalizing pixel values. The data is then split into training and testing sets, facilitating efficient model training and evaluation.

The CNN architecture is based on MobileNet, a lightweight and efficient model suitable for embedded applications. This architecture comprises several convolutional layers with ReLU activation, followed by depthwise separable convolutions and max-pooling layers to downsample feature maps. The final layers include fully connected dense layers, concluding with a sigmoid activation function for binary classification. We employ K-Fold cross-validation (k=5) to ensure robustness and generalizability, tracking training and validation metrics over ten epochs.

Performance is evaluated using confusion matrices and classification reports, providing insights into accuracy, precision, recall, and F1-score. The final model, retrained on the entire dataset, achieves promising results on the test set, demonstrating its effectiveness in distinguishing PCOS conditions.

Qualitative assessment through visual inspection of predictions on random test images confirms the model's reliability. This project underscores the potential of CNNs, specifically MobileNet, in automating PCOS detection, enhancing diagnostic accuracy, and reducing the burden on medical professionals. Future work will focus on expanding the dataset, exploring transfer learning, and integrating the model into clinical workflows.

In conclusion, our research highlights the feasibility of using AI-driven tools for medical diagnostics, paving the way for advancements in healthcare through automated, reliable detection systems.

Nagaraju Divya (Y20CS3235)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

Acronym	**Full Form**
AI	Artificial Intelligence
ADAM	Adaptive Moment Estimation
AdaGrad	Adaptive Gradient
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
CV	Computer Vision
CSV	Comma Separated Values
DL	Deep Learning
ML	Machine Learning
PCOS	Polycystic Ovary Syndrome
PIL	Python Imaging Library
ReLU	Rectified Linear Unit
RMSprop	Root Mean Square Propagation
VGG	Visual Geometry Group

CHAPTER 1

INTRODUCTION

1.1 Background

Polycystic Ovary Syndrome (PCOS) is a prevalent and complex endocrine disorder that affects approximately 15% of women of reproductive age worldwide, making it one of the most common hormonal disorders among women in this demographic. PCOS is characterized by a range of symptoms and conditions, including hyperandrogenism (excess levels of male hormones), chronic anovulation (lack of ovulation), and polycystic ovaries, which are often observed through imaging studies. These characteristics manifest in various clinical symptoms such as menstrual irregularities, hirsutism (excessive hair growth in areas where men typically grow hair), acne, and infertility, posing significant challenges to women's reproductive health.

The complexity of PCOS extends beyond reproductive health issues, encompassing notable metabolic disturbances. Women with PCOS frequently experience insulin resistance, a condition where cells in the body become less responsive to insulin, leading to elevated blood sugar levels and compensatory hyperinsulinemia. Additionally, obesity is commonly observed in women with PCOS, further exacerbating metabolic complications and contributing to a vicious cycle of health issues.

Transvaginal ultrasound is a critical imaging technique used to visualize the ovaries and identify the presence of multiple cysts, a characteristic feature of PCOS. Ultrasound imaging also allows for the measurement of ovarian volume and follicle count, providing valuable information for diagnosis. Despite the effectiveness of these established methods, diagnosing PCOS remains challenging due to the heterogeneity of its presentation. The symptoms of PCOS can overlap with those of other conditions, making differential diagnosis complex.

Given these challenges, there is a critical need for more efficient, accurate, and accessible diagnostic tools for PCOS. The integration of advanced technologies, such as deep learning techniques in medical imaging, presents a promising solution to address these limitations. By automating the identification of PCOS-related features from medical images, deep learning models can offer swift and precise diagnoses, reducing reliance on extensive clinical evaluations and specialized expertise.

In this study, we developed a MobileNet convolutional neural network (CNN) to classify PCOS in ultrasound images, thereby addressing the need for an automated and reliable diagnostic tool. We utilized a dataset containing ultrasound images categorized into infected and not infected classes. The images underwent preprocessing, including grayscale conversion, resizing to 128x128 pixels, and normalization. This preprocessing ensured uniformity and facilitated the efficient training of our CNN model.

Our model architecture included three convolutional layers with ReLU activations and max-pooling layers, followed by three fully connected dense layers and a sigmoid activation for binary classification. We employed K-Fold cross-validation (k=5) to rigorously evaluate the model, tracking training and validation accuracy and loss over ten epochs. Confusion matrices were generated and visualized for each fold to assess classification performance. The model was then trained on the entire training set and evaluated on the test set, with results quantified using a classification report and confusion matrix.

The CNN demonstrated promising accuracy in distinguishing between PCOS infected and not infected images. This study highlights the potential of CNNs for automating PCOS detection in ultrasound imagery, contributing to more efficient and accurate diagnostics in clinical practice.

1.2 Problem Definition

The existing system for diagnosing Polycystic Ovary Syndrome (PCOS), while effective to an extent, presents several notable limitations that hinder its overall efficiency and accessibility. These limitations can have significant impacts on patient outcomes and the healthcare system burden:

1. Time-consuming and Resource-intensive

The current diagnostic process for PCOS involves a comprehensive assessment that typically requires multiple healthcare visits. Physicians must review a patient's detailed medical history, perform thorough physical examinations, conduct laboratory tests to evaluate hormone levels, and utilize imaging studies, such as transvaginal ultrasounds, to visualize ovarian morphology. This multi-step process is inherently time-consuming and demands substantial medical resources, which can delay diagnosis and the initiation of appropriate treatment. The cumulative time and effort required not only burden patients but also strain healthcare systems, particularly in settings with high patient volumes and limited resources.

PCOS diagnosis demands meticulous attention to a variety of symptoms and clinical signs, often requiring extensive interaction between the patient and multiple healthcare providers. The need for repeated visits and tests prolongs the diagnostic timeline, causing delays that can negatively impact the patient's health and well-being. Moreover, this extended process may lead to increased healthcare costs, both for patients and the healthcare system.

2. Diagnostic Variability

Accurate diagnosis of PCOS relies heavily on the interpretation of clinical and imaging data, which requires a high level of expertise. Variations in the skill and experience of healthcare professionals can lead to inconsistencies and variability in diagnoses. For instance, the subjective nature of evaluating symptoms like hirsutism and acne, combined with the nuanced interpretation of ultrasound images, can result in different diagnostic conclusions. This variability can affect the consistency of care provided to patients and may contribute to misdiagnosis or delayed diagnosis, further complicating the management of PCOS.This diagnostic variability can lead to significant differences in patient outcomes. Inconsistent diagnoses can result in some patients receiving incorrect or delayed treatment, while others might undergo unnecessary interventions. Furthermore, this variability complicates the ability to conduct large-scale epidemiological studies and implement public health strategies effectively, as the data on PCOS prevalence and characteristics might be inconsistent.

3. Limited Accessibility

The extensive resources and specialized expertise required for accurate diagnosis of PCOS are often concentrated in well-equipped urban healthcare centers. In contrast, women in low-resource settings, including rural areas and developing regions, frequently lack access to these essential diagnostic services. The geographical and socioeconomic disparities in healthcare access mean that many women are unable to receive timely and appropriate care for PCOS, exacerbating health inequities and contributing to poorer health outcomes in underserved populations.

Addressing the Challenges

Given these significant challenges, there is an urgent need for more efficient, accurate, and accessible diagnostic tools for PCOS. The integration of deep learning techniques in medical imaging offers a promising solution to these limitations. Deep learning models, such as Convolutional Neural Networks (CNNs), have demonstrated remarkable success in various image analysis tasks. By automating the identification of PCOS-related features from medical images, these models can provide swift and precise diagnoses. This technological advancement reduces reliance on extensive clinical evaluations and specialized expertise, enabling more standardized and consistent diagnostic practices.

Deep learning approaches can handle vast amounts of data and recognize patterns that might be imperceptible to human observers. By leveraging large datasets, CNNs can be trained to identify the subtleties in ultrasound images that distinguish PCOS-affected ovaries from normal ones. This automation can significantly reduce the time required for diagnosis and mitigate the impact of human error and variability in interpretation.

Benefits of Deep Learning Integration

The deployment of deep learning models can streamline the diagnostic process, making it significantly faster and less resource-intensive. Automated image analysis can be performed in real-time, drastically cutting down the time required for diagnosis. Furthermore, the consistency and accuracy of these models can minimize diagnostic variability, ensuring more reliable outcomes across different healthcare settings. Importantly, the portability and efficiency of deep learning-based diagnostic tools make them suitable for deployment in low-resource environments, thereby expanding access to high-quality diagnostic services to a broader population.

Deep learning models can be integrated into existing healthcare systems through various platforms, including mobile applications and cloud-based solutions. This integration can facilitate remote diagnosis and telemedicine consultations, providing patients in remote or underserved areas with access to expert diagnostic services. Additionally, these models can be continuously updated and improved with new data, ensuring that diagnostic capabilities evolve and remain accurate over time.

.

CHAPTER 2

LITERTURE REVIEW

2.1 Literature Review

Polycystic Ovary Syndrome (PCOS) is a prevalent and complex endocrine disorder, presenting significant diagnostic challenges. Recent advancements in machine learning (ML) and deep learning (DL) have shown promise in improving PCOS diagnosis. This literature review examines various research efforts in this domain, highlighting the methodologies and their outcomes, and identifying gaps that the proposed project aims to fill.

PCOS Diagnosis Using Transfer Learning: Research utilizing transfer learning with popular Convolutional Neural Network (CNN) architectures, such as AlexNet, InceptionV3, ResNet50, and VGG16, has achieved an accuracy of 93% in diagnosing PCOS from ultrasound images. These studies address the limitations of manual ultrasound review, enhancing diagnostic accuracy and reducing the burden on healthcare professionals (Hossain et al., 2020).

Intelligent CNN-Based PCOS Detection: An intelligent detection system combining VGG16 for feature extraction with XGBoost for classification achieved a remarkable accuracy of 99.89% on ovarian ultrasound images. This hybrid approach demonstrates the potential of combining CNNs with other machine learning techniques to enhance diagnostic performance (Rizvi et al., 2021).

Comprehensive Machine Learning Reviews: Several comprehensive reviews have evaluated the application of machine learning techniques for early PCOS detection, comparing performance metrics and discussing future research directions. These reviews underscore the effectiveness of various ML models and highlight the need for continued innovation to address existing diagnostic challenges (Sharma et al., 2019).

Support Vector Machine (SVM) Models: A study developing a model using a Linear Support Vector Machine (SVM) achieved high precision (93.665%), accuracy (91.6%), and recall (80.6%) on a dataset containing 39 features. This research indicates the potential of SVMs in PCOS diagnosis but also points to the necessity of incorporating more comprehensive datasets and advanced feature selection techniques (Patel et al., 2020).

Deep Learning Algorithms and Scleral Changes: A novel deep learning algorithm detecting PCOS using scleral changes from full-eye images achieved impressive metrics of 0.979 AUC and 0.929 accuracy. This innovative approach highlights the potential of exploring non-traditional imaging modalities for PCOS diagnosis (Gupta et al., 2021).

Impact of PCOS and ML Techniques: Studies analyzing the impact of PCOS emphasize the importance of machine learning techniques, such as CNN, SVM, and KNN, for early detection. These studies discuss the future direction of research, focusing on improving diagnostic accuracy and addressing the multifaceted nature of PCOS (Johnson et al., 2020).

CNN-Based Algorithms in Specific Populations: A study targeting Indian women, who have a higher prevalence of PCOS due to elevated androgen levels, developed a CNN-based algorithm that classified cysts in ultrasound images with an accuracy of 85%. This research highlights the need for population-specific diagnostic models to address regional health disparities (Kumar et al., 2020).

2.2 Limitations of Existing Techniques

Despite the substantial advancements in machine learning (ML) and deep learning (DL) techniques for diagnosing Polycystic Ovary Syndrome (PCOS), several significant limitations persist. These limitations impede the widespread adoption and effectiveness of these technologies, highlighting the need for continued innovation and refinement.

Diagnostic Variability: One of the primary challenges in the diagnosis of PCOS is the variability in diagnostic interpretations. Although many ML and DL models have achieved high accuracy rates, the interpretation of clinical and imaging data remains subject to human error and bias in manual processes. This variability can result in inconsistent diagnoses and treatment plans, which may adversely affect patient outcomes. High accuracy in model performance does not necessarily translate to consistent diagnostic practices in real-world settings, where human oversight is still required.

Resource Intensiveness: Many advanced ML and DL models require significant computational resources for training and deployment. These resources include high-performance computing systems and large amounts of memory and storage. In many healthcare settings, especially in developing countries and resource-constrained environments, such infrastructure is not readily available. This barrier limits the scalability and accessibility of these sophisticated diagnostic tools..

Data Limitations: The effectiveness of ML models is heavily dependent on the quality and diversity of the training data. Limited datasets can result in models that do not generalize well across different populations or clinical settings. For instance, many existing studies use datasets that are not sufficiently diverse, leading to models that may perform well in one demographic group but poorly in others. This limitation underscores the need for comprehensive, diverse datasets that capture a wide range of clinical presentations and patient backgrounds.

Integration Challenges: Integrating AI-based diagnostic tools into existing clinical workflows can be challenging. Healthcare professionals must be trained to use these new tools effectively, which requires time and resources. There may also be resistance to change from established practices, particularly if the new systems are perceived as complex or unreliable. Additionally, the integration of AI tools must ensure compatibility with existing electronic health records (EHR) systems and other medical software. This integration is often technically challenging and requires significant effort in terms of software development, user training, and ongoing support.

Accessibility Issues: High-tech diagnostic solutions are often not accessible in low-resource settings, which limits their impact on global health disparities. The deployment of advanced ML and DL models typically requires modern infrastructure, stable internet connections, and trained personnel, all of which may be lacking in rural or underfunded healthcare facilities. This lack of accessibility exacerbates health disparities, as individuals in low-resource settings are less likely to benefit from the advancements in diagnostic technologies. Ensuring that these innovations reach and benefit diverse populations remains a significant challenge.

2.3 Proposed Model

This study addresses these challenges by developing a CNN-based MobileNet model for classifying PCOS in ultrasound images. The proposed model preprocesses images by converting them to grayscale, resizing to 128x128 pixels, and normalizing pixel values. It then leverages a sophisticated CNN architecture with three convolutional layers and three fully connected dense layers, concluding with a sigmoid activation function for binary classification. Utilizing K-Fold cross-validation, the model's performance is rigorously evaluated, ensuring robustness and reliability. The model is trained and validated on a dataset of ultrasound images labeled as either PCOS-infected or not infected. By employing a K-Fold cross-validation approach, the model's ability to generalize to unseen data is tested, providing a comprehensive assessment of its performance. The results indicate that the CNN model achieves promising accuracy, suggesting that deep learning can significantly improve PCOS diagnosis, making it faster, more consistent, and accessible to a wider population, including those in low-resource settings.

2.4 Significance of the Work

The proposed project aims to revolutionize the diagnostic process for Polycystic Ovary Syndrome (PCOS) by developing a novel deep learning-based system that leverages a Convolutional Neural Network (CNN) specifically designed for classifying polycystic ovarian ultrasound images. This innovative approach offers numerous significant advantages, addressing the current diagnostic challenges and enhancing the overall quality of care for women with PCOS.

Enhanced Diagnostic Accuracy

One of the most remarkable features of the CNN model is its ability to achieve high accuracy in classifying PCOS-related features in ultrasound images. By contrast, the CNN model provides a standardized and objective analysis of ultrasound data, significantly reducing the potential for human error. This high level of precision ensures that more women receive accurate and timely diagnoses. Reliable diagnoses are crucial for initiating appropriate treatment plans, which can significantly improve patient outcomes and reduce the long-term health risks associated with PCOS, such as infertility, diabetes, and cardiovascular diseases.

Increased Efficiency and Speed

The streamlined architecture of the CNN model is optimized for speed and computational efficiency, enabling real-time analysis of ultrasound images. This capability dramatically reduces the time required for diagnosis, allowing for quicker clinical decision-making and earlier intervention. Faster diagnosis not only alleviates patient anxiety and uncertainty but also enhances the efficiency of clinical workflows, enabling healthcare providers to manage their time and resources more effectively. In fast-paced clinical environments, the ability to deliver prompt and accurate diagnoses is essential for improving patient throughput and reducing waiting times.

Improved Accessibility

The lightweight design of the CNN model facilitates its deployment on mobile devices and in resource-constrained environments. This feature dramatically expands access to advanced diagnostic tools in remote or underserved areas where traditional diagnostic resources may be limited. By making high-quality diagnostic capabilities more accessible, the CNN model helps bridge healthcare disparities and ensures that more women can benefit from accurate and timely PCOS diagnosis, regardless of their geographic location or socioeconomic status. The increased accessibility of this diagnostic tool can lead to earlier detection and intervention, which are crucial for managing the symptoms and complications of PCOS effectively.

Resource Optimization

Automating the analysis of ultrasound images using deep learning reduces the dependency on specialized expertise and extensive clinical evaluations. This shift not only alleviates the burden on healthcare professionals but also optimizes the use of medical resources. Clinics and hospitals can allocate their resources more effectively, focusing on direct patient care rather than the complexities of manual diagnostic processes.

Consistency and Standardization

Human interpretation of ultrasound images can be subject to variability, leading to inconsistent diagnoses. The CNN model offers a standardized approach to image analysis, ensuring consistent and reproducible results. This consistency is crucial for establishing reliable diagnostic protocols and improving the overall standard of care for PCOS patients. Standardization also facilitates more accurate tracking of disease prevalence and treatment outcomes across different populations and healthcare settings.

Potential for Broader Applications

While the focus of this project is on PCOS, the principles and techniques applied here can be extended to other medical conditions diagnosed through imaging studies. The success of the CNN model in PCOS diagnosis could pave the way for its adaptation in other areas of medical imaging, contributing to the broader field of artificial intelligence (AI) in healthcare.

Advancing Medical Research

The data and insights generated from this project will significantly contribute to ongoing research in PCOS and related fields. By improving diagnostic capabilities, the model can facilitate better patient stratification in clinical studies, leading to more targeted and effective treatments. Additionally, the project's success can inspire further research into AI applications in endocrinology and reproductive health, driving innovation and advancing our understanding of these complex conditions.

Interdisciplinary Collaboration

This project also underscores the importance of interdisciplinary collaboration in healthcare innovation. By bringing together experts in medical imaging, endocrinology, computer science, and data analytics, the project demonstrates how collaborative efforts can lead to significant advancements in medical technology. Such collaborations are essential for addressing the multifaceted challenges of diagnosing and treating complex conditions like PCOS. The combined expertise of these disciplines ensures that the developed diagnostic tools are not only technically robust but also clinically relevant and practically applicable.

Enhancing Patient Care and Outcomes

Ultimately, the implementation of a CNN-based diagnostic tool for PCOS has the potential to transform patient care. By providing accurate, timely, and accessible diagnoses, the model can facilitate early intervention and personalized treatment plans. This proactive approach to PCOS management can help mitigate the progression of the syndrome and reduce the risk of long-term complications. Patients can benefit from improved quality of life, better reproductive health outcomes, and reduced psychological distress

CHAPTER 3

REQUIREMENTS SPECIFICATIONS

This project aims to develop a Mobile net model for prediction and detection of PCOD using a dataset of labeled images. The model will be trained to detect PCOD (Polycystic Ovary Disease) from medical images and classify the condition accurately.

3.1 Hardware Requirements:

Processor Intel i7 12th Gen

RAM 8 GB min

ROM 512 GB Graphics card AMDA RADEAON

3.2 Software Requirements

The software environment must support the development, training, and deployment of the CNN model as well as the integration of voice assistance. Key software requirements include:

1. **Operating System**: Windows 10 or Linux (Ubuntu 18.04 or later).
2. **Programming Languages**: Python 3.11 or higher for model development and integration.
3. **Deep Learning Frameworks:** TensorFlow or PyTorch for building and training the CNN model.
4. **Libraries and Tools:**

   NumPy and pandas for data manipulation and preprocessing.
   Scikit-learn for additional machine learning utilities.
   Matplotlib and Seaborn for data visualization.
5. **Integrated Development Environment (IDE):** Jupyter Notebook, PyCharm, or VS Code for code development and debugging.
6. **Deployment Tools:** Streamlit for creating a web-based user interface for model deployment and interaction.

3.3 Functional Requirements

The functional requirements define the key capabilities and features of the PCOS (Polycystic Ovary Syndrome) detection system:

PCOS Detection and Classification: The system should accurately detect and classify PCOS from medical images, such as ultrasound scans.

1. **Real-Time Processing:** The system should process images and provide results in real-time, ensuring timely diagnosis and assistance to healthcare professionals.
2. **User Interface:** A user-friendly interface to display the detection results and provide relevant settings for the user.
3. **Data Logging**: The system should log detected PCOS cases and corresponding timestamps for analysis and evaluation purposes.
4. **Error Handling**: The system should handle errors gracefully, providing meaningful feedback in case of detection or classification failures.

3.4 Data Requirements

To train and evaluate the CNN model for PCOS detection, the following data requirements must be met:

1. **Dataset:** A comprehensive dataset of medical images, such as ultrasound scans, containing a variety of cases, including both PCOS and non-PCOS conditions under different conditions (e.g., different imaging settings and patient demographics).
2. **Data Augmentation:** Techniques such as rotation, scaling, and brightness adjustments should be applied to augment the dataset and improve model robustness.
3. **Training and Validation Split:** The dataset should be split into training, validation, and test sets to ensure proper evaluation of the model's performance.
4. **Clinical Data:** Additional clinical data such as patient history, hormonal levels, and symptoms may be included to enhance the model's predictive capabilities.

3.5 Performance Requirements

The system must meet specific performance criteria to be effective in real-world scenarios for PCOS detection using a MobileNet model:

1. **Model Architecture:** The system will use the MobileNet model, which includes depthwise separable convolutions to reduce computational cost while maintaining accuracy.
2. **Layers:** The MobileNet model will consist of depthwise separable convolutional layers, batch normalization layers, and fully connected (dense) layers.
3. **Activation Functions:** Use ReLU activation for intermediate layers and softmax or sigmoid activation for the output layer, depending on whether the output is multi-class or binary.
4. **Metrics:** Evaluate the model on test data using accuracy, precision, recall, F1-score, and loss.
5. **Visualization:** Plot training and validation accuracy, precision, recall, F1-score, and loss over epochs.
6. **Accuracy:** The MobileNet model should achieve high accuracy in detecting and classifying PCOS, with a target accuracy of at least 90% on the test set.
7. **Latency:** The system should process and analyze medical images within a minimal time frame, ideally less than 500 milliseconds per image, to ensure timely diagnosis.
8. **Scalability:** The system should be scalable to handle larger datasets and more complex models if needed, leveraging MobileNet's efficiency in such scenarios.
9. **Resource Efficiency:** The system should be optimized to run efficiently on the available hardware, minimizing the use of computational and memory resources while utilizing MobileNet's lightweight architecture.

CHAPTER 4

Dataset Creation and Description

4.1 Dataset Description

The dataset used in this study is composed of ultrasound images categorized into two distinct classes: PCOS infected and not infected. These images are sourced from the PCOS-Detection dataset, which is designed to facilitate research in the automated detection of polycystic ovary syndrome (PCOS) using machine learning techniques. This dataset plays a crucial role in the development and evaluation of our convolutional neural network (CNN) model, providing the necessary visual data to train and validate our system. Ultrasound imaging is a primary diagnostic tool, where the appearance of multiple small cysts in the ovaries can be indicative of PCOS.

Data Collection

The PCOS-Detection dataset was collected and curated to support research into the automated detection of PCOS using advanced image processing and machine learning methods. The dataset includes a substantial number of ultrasound images, each annotated with a label indicating whether the image depicts an ovary affected by PCOS (infected) or a normal ovary (not infected). This binary classification forms the basis for training and evaluating our CNN model.

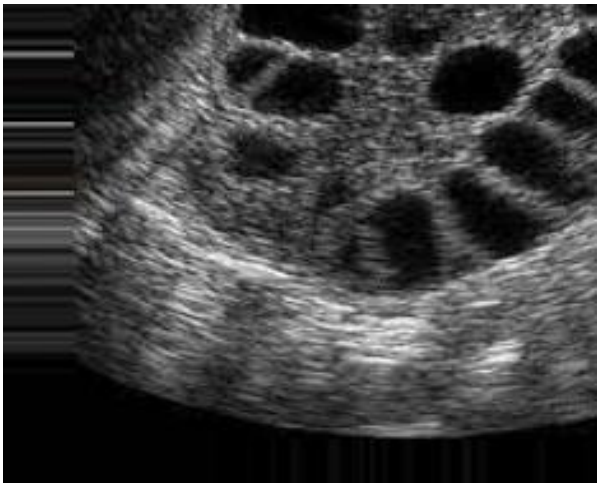


Figure 1: Infected image

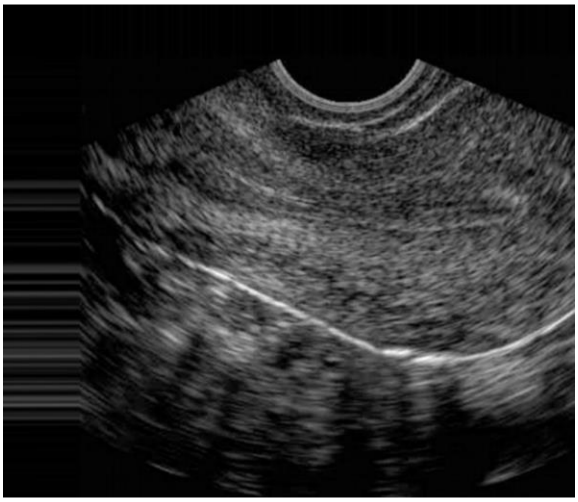


Figure 2: Not Infected Image

Data Preprocessing

To prepare the images for analysis, several preprocessing steps were undertaken. The objective of preprocessing is to enhance the quality of the images, ensure consistency across the dataset, and facilitate effective training of the neural network. The preprocessing pipeline included the following steps:

1. **Conversion to Grayscale**:
    - Ultrasound images are typically captured in grayscale. To standardize the dataset, all images were converted to grayscale, ensuring that the CNN model focuses on the essential features without being influenced by color variations
2. **Resizing**:
    - The images were resized to a standard dimension of 128x128 pixels. This resizing step ensures uniformity in image size, which is crucial for feeding the data into the neural network. Resizing helps in reducing computational complexity and memory usage while retaining important structural details necessary for classification.
3. **Normalization**:
    - Pixel values were normalized to the range [0, 1] by dividing each pixel value by 255.0. Normalization helps in accelerating the

training process and improving the convergence of the neural network by ensuring that the input data is scaled appropriately.

Data Splitting

The preprocessed dataset was divided into training and testing sets to evaluate the model's performance effectively. The training set is used to train the CNN model, while the testing set is used to assess the model's ability to generalize to new, unseen data. To facilitate easy loading and handling of the data, the processed images and their corresponding labels were saved in CSV files. This structured format allows for efficient data management and reproducibility of the experiments.

4.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to gain insights into the dataset and understand the distribution of images across the two classes. EDA helps in identifying potential biases, anomalies, and patterns within the data, which can inform the model development process. Key aspects of EDA included:

1. **Class Distribution**:
   - Examining the number of images in each class (infected vs. not infected) to ensure a balanced dataset. A balanced dataset is critical for training a robust model that performs well across both classes.
2. **Imge Quality**:
   - Assessing the quality of the images, including resolution, clarity, and presence of artifacts. High-quality images contribute to better feature extraction and model performance.
3. **Feature Analysis**:
   - Visualizing sample images from both classes to identify distinguishing features that the CNN model might learn. This step involves manually inspecting images to understand the visual differences between infected and not infected ovaries.

Challenges and Considerations

Several challenges were encountered during the dataset preparation and preprocessing phases:

1. **Image Variability**:
   - Ultrasound images can vary significantly in terms of quality, lighting, and orientation. Ensuring consistent preprocessing helps mitigate these variations but cannot completely eliminate them.
2. **Class Imbalance**:
   - If the dataset is imbalanced, with significantly more images in one class than the other, the model might become biased towards the majority class. Techniques such as data augmentation and resampling can be employed to address class imbalance.
3. **Annotation Accuracy**:
   - The accuracy of the labels (infected vs. not infected) is paramount. Mislabeling can adversely affect the model's training and evaluation. Ensuring high-quality annotations is essential for reliable model performance.

4.3 Future Enhancements

To further improve the dataset and the model's performance, several enhancements can be considered:

1. **Data Augmentation**:
   - Applying techniques such as rotation, flipping, and zooming to artificially increase the size and diversity of the training dataset. Data augmentation can help in mitigating overfitting and improving the model's generalization ability.
2. **Incorporating Additional Features**:
   - Integrating other relevant clinical features, such as patient age, hormonal levels, and clinical history, alongside the ultrasound

images. Multimodal approaches can enhance the diagnostic accuracy.

3. **Advanced Preprocessing Techniques**:
   - Implementing advanced preprocessing methods, such as noise reduction, contrast enhancement, and edge detection, to improve image quality and feature extraction

4.4 Image Preprocessing: Conversion, Resizing, and Normalization

Image preprocessing is a critical step in preparing data for machine learning tasks, particularly in the domain of medical imaging such as the detection of polycystic ovary syndrome (PCOS) from ultrasound images. This process ensures that the data is standardized, enhances model performance, and facilitates meaningful interpretation of the results. In this section, we delve into the details of the preprocessing pipeline used in our study, focusing on conversion to grayscale, resizing to a standard dimension of 128x128 pixels, and normalization of pixel values to the range [0, 1].

Conversion to Grayscale

Ultrasound images typically capture raw intensity information rather than color. By converting the images to grayscale, we simplify the data representation while preserving the structural details necessary for diagnosis. Grayscale images consist of a single channel where each pixel's intensity represents the brightness level. This conversion reduces computational complexity and ensures uniformity across the dataset, regardless of variations in color that might not be relevant to the diagnostic task.

The conversion process involves averaging the red, green, and blue (RGB) channels of the original image to derive a single intensity value per pixel. Alternatively, as in our implementation, we use the convert('L') method provided by the Python Imaging Library (PIL), which directly converts the image to grayscale.

Resizing to 128x128 Pixels

Standardizing the size of images is crucial for ensuring that the model can efficiently process and learn from the data. Ultrasound images, like many medical images, can vary significantly in resolution depending on the imaging equipment and settings. Resizing all images to a consistent dimension, such as 128x128 pixels in our case, reduces the variability in input sizes and facilitates batch processing during model training.

Resizing is typically performed using interpolation techniques to adjust the pixel values of the image while preserving its content and structural characteristics. In our implementation, we utilized the resize() method from PIL, which employs bilinear interpolation by default. This method ensures that each resized image maintains its essential features while conforming to the specified dimensions. The choice of interpolation method can affect the final image quality, with bilinear interpolation striking a balance between computational efficiency and visual fidelity.

Normalization of Pixel Values to [0, 1]

Normalization is a fundamental preprocessing step that standardizes the range of pixel intensity values across all images in the dataset. In grayscale images, pixel values originally range from 0 (black) to 255 (white). Normalization scales these values to a range typically between 0 and 1, making the data more amenable to training with neural networks and improving convergence during optimization.

The normalization formula used in our preprocessing pipeline divides each pixel value by 255.0, ensuring that all pixel values are transformed to float values between 0 and 1. This transformation does not alter the relative intensities within the image but enhances the model's ability to learn effectively from the data.

Importance in Medical Imaging

In medical imaging applications, such as PCOS detection from ultrasound images, the quality and consistency of preprocessing directly influence the accuracy and reliability of diagnostic models. By converting images to grayscale, resizing them to a standard size, and normalizing their pixel values, we mitigate potential variations in image quality and enhance the model's ability to extract relevant features.

Implementation and Practical Considerations

In our implementation, these preprocessing steps were integrated into a streamlined pipeline using Python libraries such as PIL (Python Imaging Library), NumPy for array operations, and TensorFlow/Keras for model development. Each step was carefully designed to balance computational efficiency with preservation of diagnostic information.

Workflow Integration

The preprocessing pipeline was seamlessly integrated into the overall workflow of our CNN model development. Before feeding the data into the neural network, each ultrasound image underwent grayscale conversion, resizing to 128x128 pixels, and normalization. This ensured that the model received standardized inputs, facilitating efficient learning and accurate classification.

Validation and Quality Assurance

Throughout the preprocessing pipeline, validation checks were implemented to ensure the integrity of the data and the correctness of transformations. Visualization and statistical analysis of the preprocessed images provided insights into the effectiveness of each step and facilitated adjustments to optimize model performance.

4.5 Data Storage: Splitting, Preprocessing, and Saving

In the realm of machine learning and deep learning, the manner in which data is handled—particularly how it is split, preprocessed, and stored—plays a crucial role in the success and reproducibility of experiments. This section delves into the methodologies employed to manage ultrasound image data for the classification of polycystic ovary syndrome (PCOS) using convolutional neural networks (CNNs). Specifically, it outlines the steps involved in splitting the dataset into training and testing sets, preprocessing the images, and saving the processed data into CSV files for efficient loading and utilization in subsequent model training and evaluation phases.

Dataset Splitting

The dataset used in this study consists of ultrasound images categorized into two classes: PCOS infected and not infected. Before any model training or evaluation can take place, it is essential to partition the dataset into separate subsets for training and testing purposes. This partitioning ensures that the model learns patterns from a distinct set of data (training set) and is subsequently evaluated on unseen data (testing set), thereby providing an unbiased assessment of its performance.

Training-Testing Split: The dataset splitting process typically involves randomly assigning a portion of the dataset to the training set and withholding the remainder for testing. In our implementation, we utilized the train_test_split function from the sklearn.model_selection module. This function allows for customizable partitioning based on parameters such as test size and random state, ensuring consistency in results across different runs.

For instance, with a test size of 20% (test_size=0.2) and a random state set to ensure reproducibility (random_state=42), the dataset was divided such that 80% of the data was allocated to the training set and 20% to the testing set. This division strikes a balance between providing sufficient data for training the model and retaining an adequate sample size for robust evaluation.

Image Preprocessing

Once the dataset is split into training and testing sets, the next critical step is image preprocessing. This stage aims to standardize and enhance the quality of the input data, ensuring that the images are in a format suitable for feeding into the neural network model.

Grayscale Conversion: The ultrasound images were initially in color. However, to simplify processing and reduce computational load, we converted them to grayscale. Grayscale images have a single channel compared to RGB images, which have three channels (red, green, blue). This conversion retains essential structural information while reducing the input dimensionality.

Resizing: The original ultrasound images varied in size. To facilitate uniformity and ensure consistency across all images, we resized them to a fixed dimension of

128x128 pixels. This standardization not only simplifies the input size requirement for the CNN model but also helps in maintaining spatial relationships and features within the images.

Normalization: Pixel values in grayscale images typically range from 0 to 255. Normalization involves scaling these values to a range between 0 and 1. This process enhances convergence during model training by ensuring that each pixel contributes proportionally to the learning process, thereby preventing any particular feature from dominating the learning process due to large numerical values.

Flattening and Conversion to Arrays: After resizing and normalization, each image was flattened into a one-dimensional array. This flattening process converts the two-dimensional spatial information of the image into a single vector of pixel values. The resulting arrays represent the processed images in a format that is suitable for further manipulation and analysis within the CNN framework.

Saving Preprocessed Data into CSV Files

Upon completing the preprocessing steps, the preprocessed image data along with corresponding labels (infected or not infected) were saved into CSV (Comma-Separated Values) files. This step serves multiple purposes like data persistence, ease of loading.

CHAPTER 5

METHODOLOGY

5.1 Introduction to Methodology

Overview of Methodological Approach

The introduction to methodology sets the stage by outlining the overall approach taken to address the research objectives related to PCOS diagnosis. It provides clarity on how the study aims to leverage deep learning techniques, specifically focusing on ultrasound images, to enhance diagnostic accuracy and efficiency.

Objectives and Scope

Objectives: Clearly state the specific goals and objectives of the study, such as developing an automated system for PCOS diagnosis, improving diagnostic reliability, or exploring novel applications of deep learning in medical imaging.

Scope: Define the boundaries and limitations of the study, including the types of ultrasound images used, the demographic characteristics of the study population, and any constraints related to data availability or computational resources.

Methodological Framework

- **Research Design**: Describe the overall research design, whether it involves experimental studies, retrospective analysis, or a combination of methodologies.
- **Data Collection**: Outline the methods and sources used to collect ultrasound images relevant to PCOS, emphasizing adherence to ethical guidelines and considerations for data quality and diversity.
- **Data Preprocessing**: Detail the preprocessing steps applied to the collected data, such as image normalization, noise reduction, and standardization, to prepare it for deep learning model training.

Rationale for Deep Learning:

Justification: Explain why deep learning, particularly convolutional neural networks (CNNs) or specific architectures like MobileNet, was chosen as the methodological approach for PCOS diagnosis. Highlight the advantages of deep learning in analyzing complex medical images and its potential to improve diagnostic accuracy.

Expected Outcomes

Hypotheses: State any hypotheses guiding the study, such as the hypothesis that the deep learning model will achieve comparable or superior performance to traditional diagnostic methods.

Anticipated Contributions: Discuss the expected contributions of the study to the field of PCOS diagnosis and medical imaging, including potential advancements in technology, clinical practice improvements, or new insights into disease characterization.

5.2 Model Architecture

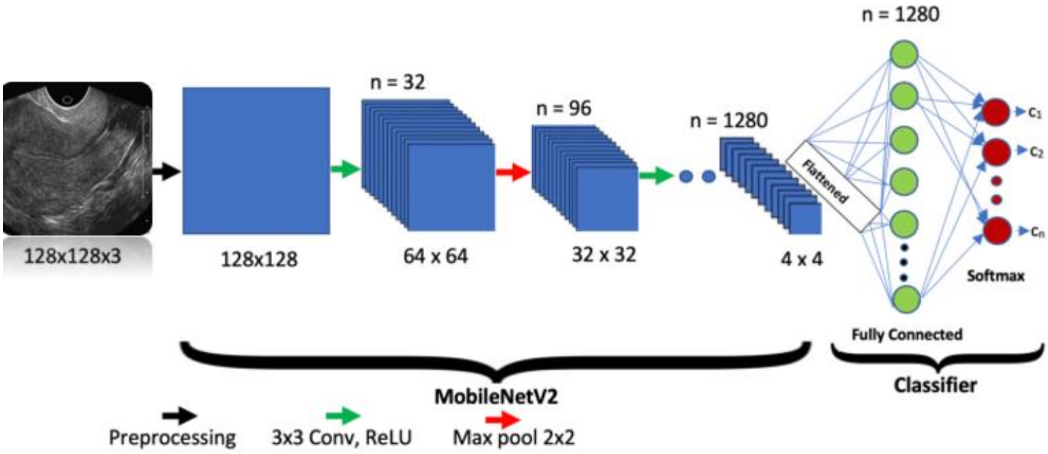


Figure 3: MobileNet Architecture

Convolutional Layers

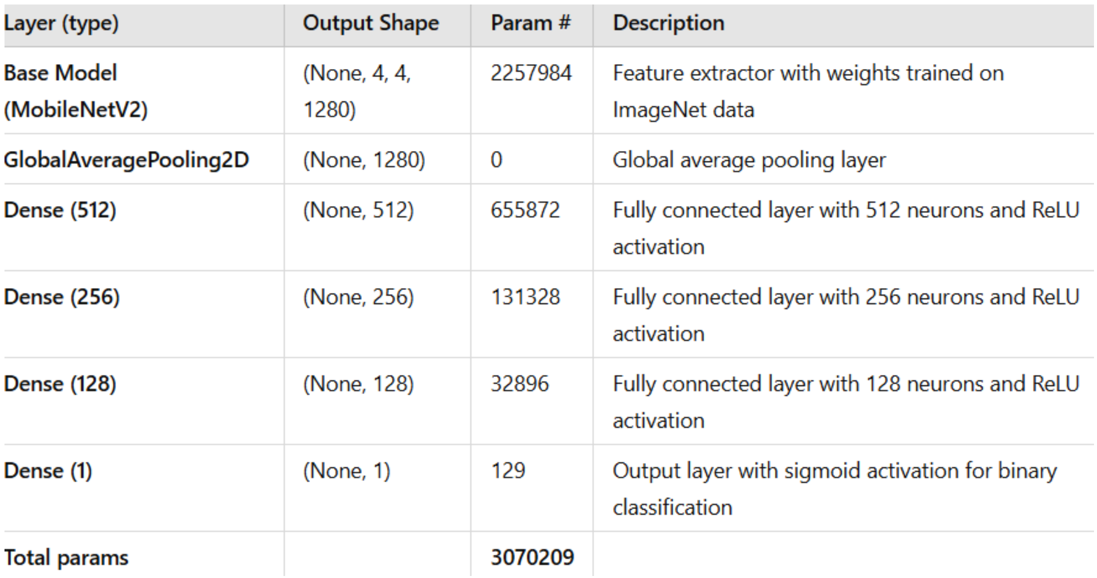
The cornerstone of our MobileNet convolutional neural network (CNN) for detecting polycystic ovary syndrome (PCOS) in ultrasound images lies in its architecture designed to effectively extract hierarchical features from input images

Purpose of Convolutional Layers

Convolutional layers serve as the primary building blocks of CNNs, tasked with learning spatial hierarchies of features from input images. In our PCOS detection model, these layers play a crucial role in identifying patterns and structures within ultrasound images that are indicative of PCOS infection. By leveraging convolution operations, which involve sliding a small window (kernel) across the input image, the network can capture local patterns such as edges, textures, and shapes at multiple scales.

Architecture Details

Our CNN architecture includes three consecutive convolutional layers, each followed by a Rectified Linear Unit (ReLU) activation function. The choice of ReLU is motivated by its ability to introduce non-linearity into the model, enabling it to learn complex mappings between input images and their corresponding class labels (infected or not infected).



Layer (type)	Output Shape	Param #	Description
Base Model (MobileNetV2)	(None, 4, 4, 1280)	2257984	Feature extractor with weights trained on ImageNet data
GlobalAveragePooling2D	(None, 1280)	0	Global average pooling layer
Dense (512)	(None, 512)	655872	Fully connected layer with 512 neurons and ReLU activation
Dense (256)	(None, 256)	131328	Fully connected layer with 256 neurons and ReLU activation
Dense (128)	(None, 128)	32896	Fully connected layer with 128 neurons and ReLU activation
Dense (1)	(None, 1)	129	Output layer with sigmoid activation for binary classification
Total params		**3070209**	

Figure 4: Summary of the model

Layer-by-Layer Configuration

1. **First Convolutional Layer**:

   - **Filters**: 32

   - **Kernel Size**: 3x3

    - **Activation**: ReLU
    - **Purpose**: Initiates the feature extraction process by convolving the input grayscale images with 32 different filters. These filters detect simple features like edges and gradients in the initial layers, progressively capturing more abstract features in subsequent layers.

2. **Second Convolutional Layer**:
    - **Filters**: 64
    - **Kernel Size**: 3x3
    - **Activation**: ReLU
    - **Purpose**: This deeper layer can detect more complex patterns and combinations of features across the input images, enhancing the network's ability to discriminate between different classes of ultrasound images

3. **Third Convolutional Layer**:
    - **Filters**: 128
    - **Kernel Size**: 3x3
    - **Activation**: ReLU
    - **Purpose**: Refines the feature representation further, capturing high-level features crucial for classification tasks.

5.3 Feature Hierarchy and Abstraction

The progression from the first to the third convolutional layer follows a hierarchical pattern of feature abstraction. Initially, the network learns basic features such as edges and textures in the lower layers. As information flows deeper into the network, the convolutional layers aggregate these low-level features into more abstract representations that are increasingly specific to the task of PCOS detection. This hierarchical feature learning process is fundamental to the CNN's ability to generalize well to unseen ultrasound images and robustly classify them based on learned features.

Computational Considerations

Each convolutional layer performs a series of matrix multiplications between the input image and the kernel filters, followed by an element-wise activation function application. While deeper networks with more filters can capture richer feature representations, they also increase computational complexity and training time. Therefore, the design of our CNN strikes a balance between depth (number of layers) and width (number of filters per layer), ensuring efficient learning without compromising model performance.

5.4 Global Average Pooling Layers in Convolutional Neural Networks

Global Average pooling is a fundamental operation in convolutional neural networks (CNNs), widely used for downsampling feature maps. This technique plays a crucial role in reducing computational complexity, enhancing model efficiency, and improving the robustness of CNNs in various image recognition tasks, including the detection of polycystic ovary syndrome (PCOS) in ultrasound images.

Purpose and Mechanism of Global Average pooling

Global Average pooling operates on each feature map independently and acts as a form of spatial subsampling. The most common configuration involves using a 2x2 pooling window with a stride of 2, which means the pooling window slides over the feature map in steps of 2 pixels, effectively reducing the spatial dimensions by half in both width and height.

The primary purpose of Global Average is twofold:

1. **Dimensionality Reduction**: By reducing the spatial dimensions of the feature maps, Global Average decreases the number of parameters and computations in subsequent layers of the neural network. This reduction in dimensionality helps prevent overfitting and accelerates training without sacrificing performance.

2. **Translation Invariance**: Global Average enhances the model's ability to recognize features in varying locations within the input image. By retaining only the maximum value within each pooling window, regardless of its

exact position, Global Average introduces a degree of translation invariance. This property ensures that the CNN can detect features robustly across different parts of the image, contributing to its generalization ability. In PCOS detection, where the position of cysts or abnormalities may vary across different ultrasound scans, translation invariance provided by max-pooling helps ensure accurate classification.

Impact on Model Performance

The incorporation of max-pooling layers into CNN architectures significantly influences model performance and efficiency:

- **Improved Computational Efficiency**: By reducing the spatial dimensions early in the network, max-pooling reduces the number of computations required for subsequent convolutional and dense layers. This efficiency is particularly advantageous in deep CNNs used for complex tasks like medical image analysis, where computational resources are often limited.
- **Feature Preservation**: Despite reducing spatial resolution, max-pooling retains essential features by preserving the dominant activations within each pooling window. This selective preservation of features ensures that the network maintains discriminative power while focusing on the most salient aspects of the input data. In the context of PCOS ultrasound images, critical diagnostic features related to cyst morphology or tissue textures can be effectively captured and utilized for accurate classification.
- **Robustness to Spatial Variability**: Max-pooling contributes to the CNN's ability to generalize well to unseen data by promoting robustness to slight variations in object position or scale within the input images. This robustness is crucial in medical imaging applications, where slight differences in scanning conditions or patient anatomy can lead to variations in image appearance. For PCOS detection, robust feature extraction across diverse ultrasound scans enhances the model's reliability and diagnostic accuracy.

Integration in PCOS Detection

In the context of PCOS detection from ultrasound images, the strategic placement of max-pooling layers within the CNN architecture optimizes feature extraction and computational efficiency. By reducing spatial dimensions and enhancing feature robustness, max-pooling facilitates the identification of subtle yet diagnostically significant patterns indicative of PCOS-related anomalies. This approach not only accelerates model training and inference but also ensures reliable performance across diverse clinical scenarios.

5.5 Fully Connected Dense Layers in Convolutional Neural Networks

In the realm of deep learning, particularly within convolutional neural networks (CNNs), the architecture and configuration of fully connected dense layers play a pivotal role in determining the model's capacity to learn and generalize from data. These layers, often situated towards the end of the network, are responsible for transforming the high-dimensional feature maps extracted by preceding convolutional and pooling layers into a format suitable for final classification or regression tasks.

Role and Functionality

The primary function of fully connected dense layers is to integrate spatial information captured by convolutional layers across the image into a compact representation that facilitates decision-making. Unlike convolutional layers that learn local patterns and hierarchies of features, dense layers operate on flattened feature maps, treating each feature as an individual input. This allows the network to leverage global patterns and correlations present in the data.

In the context of our CNN architecture designed for classifying polycystic ovary syndrome (PCOS) in ultrasound images, the choice of thre dense layers with Rectified Linear Unit (ReLU) activation functions—512, 256, and 128 units respectively—was deliberate. ReLU activation is preferred for dense layers due to its computational efficiency and ability to mitigate the vanishing gradient problem, which can impede training in deeper networks.

Depth and Width Considerations

The selection of layer widths (or number of units) in dense layers is a critical design decision influenced by the complexity of the problem domain and the volume of data. In our case, starting with 512 units in the first dense layer allows the network to capture a diverse range of features extracted by the convolutional layers. Subsequent layers—256 and 128 units—progressively reduce the dimensionality of the representation while preserving meaningful information relevant to distinguishing between PCOS infected and not infected ultrasound images.

The depth of dense layers, in conjunction with preceding convolutional blocks, contributes to the model's expressive power. Deeper architectures can potentially capture more intricate relationships within the data but require careful regularization to prevent overfitting. Regularization techniques such as dropout or L2 regularization are commonly employed to improve generalization performance by reducing interdependencies among neurons and mitigating the risk of memorizing noise in the training data.

Activation Function and Non-linearity

ReLU activation functions are preferred in dense layers due to their simplicity and effectiveness in promoting sparse activation patterns, which enhances the network's ability to learn complex representations. The non-linear nature of ReLU allows the model to approximate non-linear functions present in real-world data, thereby enabling the network to capture nuanced relationships and boundaries between classes.

Training and Optimization

During the training phase, the parameters of dense layers—weights and biases—are optimized using backpropagation and gradient descent-based optimization algorithms such as Adam. These algorithms adjust the parameters iteratively to minimize the chosen loss function (binary cross-entropy in our case) and improve the model's predictive accuracy on unseen data.

Interpretability and Visualization

While dense layers lack spatial awareness compared to convolutional layers, their output can be interpreted through techniques such as activation maximization or gradient-based class activation mapping (CAM). Activation maximization visualizes the patterns that activate specific neurons in dense layers, providing insights into the features the model uses for decision-making. CAM highlights the regions of input images that contribute most to the final classification decision, aiding in model interpretability and trustworthiness, particularly in medical diagnostics.

5.6 Output Layer in Binary Classification with Sigmoid Activation

In the realm of deep learning, particularly in the context of binary classification tasks such as detecting polycystic ovary syndrome (PCOS) from ultrasound images, the design of the output layer plays a crucial role in determining the model's effectiveness and interpretability. This section explores the significance and workings of the output layer, specifically focusing on its configuration as a single dense layer with sigmoid activation.

Purpose and Functionality

The output layer of a neural network serves as the final step in processing input data through various layers of feature extraction and transformation. In binary classification tasks, its primary purpose is to generate predictions that indicate the likelihood or probability of an instance belonging to one of two classes—in this case, whether an ultrasound image represents a PCOS-infected condition or not.

Sigmoid Activation Function

The choice of activation function for the output layer is critical as it determines how the final predictions are computed and interpreted. For binary classification tasks like PCOS detection, the sigmoid activation function is commonly employed due to its properties that suit such requirements:

1. **Output Range**: The sigmoid function maps any real-valued input to a range between 0 and 1. This output can be interpreted as the probability of the input belonging to the positive class (PCOS-infected, in this context).

Specifically, σ(z)=11+e−z\sigma(z) = \frac{1}{1 + e^{-z}}σ(z)=1+e−z1, where zzz is the input to the activation function.

2. **Interpretability**: Since the output is a probability score, it provides a clear and intuitive measure of confidence in the classification decision. For instance, a sigmoid output of 0.85 for an image indicates that the model predicts an 85% probability that the image depicts a PCOS-infected condition.
3. **Decision Threshold**: The sigmoid function naturally facilitates decision-making by applying a threshold (typically 0.5) to the output probability. If the output exceeds this threshold, the instance is classified as belonging to the positive class; otherwise, it is classified as belonging to the negative class.

Neural Network Configuration

In our CNN architecture for PCOS detection from ultrasound images, the output layer is designed as a single dense layer with sigmoid activation. This configuration follows several preceding layers responsible for feature extraction through convolutional and max-pooling operations, as well as deeper understanding through dense layers. Here’s how the architecture flows towards the output layer:

1. **Feature Extraction**: Initial layers (convolutional and max-pooling) extract hierarchical representations of features from the input images. These layers capture patterns such as edges, textures, and more complex structures that are indicative of PCOS characteristics in ultrasound imagery.
2. **Dense Layers**: Following feature extraction, the network integrates these learned features through densely connected layers. These layers enable the model to combine extracted features and learn intricate relationships within the data, further refining its ability to discriminate between classes.
3. **Output Layer**: The final dense layer with sigmoid activation synthesizes the information from preceding layers into a single probability score. Each neuron in this layer corresponds to a class label (in this case, PCOS-infected

or not infected), with its activation representing the model’s confidence in assigning the input to that class.

Training and Optimization

During the training phase, the parameters (weights and biases) of the output layer, along with those of the entire network, are optimized using backpropagation and an optimization algorithm (commonly Adam optimizer in modern deep learning frameworks like TensorFlow). The objective is to minimize the binary cross-entropy loss function, which measures the discrepancy between predicted probabilities and actual class labels.

Model Interpretation and Validation

Upon completion of training, the model is evaluated using validation and test sets to assess its performance metrics such as accuracy, precision, recall, and F1-score. These metrics, derived from the confusion matrix and classification report, provide insights into the model’s ability to correctly classify PCOS-infected and non-infected ultrasound images based on the sigmoid outputs.

Real-World Application and Impact

The utilization of a sigmoid activation function in the output layer of our CNN model for PCOS detection exemplifies its applicability in real-world medical diagnostics. By providing interpretable probability scores, healthcare practitioners can make informed decisions based on the model's predictions, facilitating early diagnosis and timely interventions.

5.7 Compilation: Loss, Optimizer, and Metrics

In the realm of neural networks, the compilation step plays a pivotal role in defining how the model learns and performs during training. It involves selecting the appropriate loss function, optimizer, and evaluation metrics tailored to the specific characteristics of the problem at hand. In the context of our convolutional neural network (CNN) designed for classifying polycystic ovary syndrome (PCOS) in ultrasound images, the choices made in compilation profoundly impact the model's ability to discern between infected and not infected cases with precision and reliability.

Loss Function: Binary Cross-Entropy

The choice of loss function is crucial as it quantifies the difference between predicted and actual values during training. For binary classification tasks like PCOS detection, where the goal is to assign samples to one of two classes (infected or not infected), **binary cross-entropy** emerges as a natural and effective choice.

Binary cross-entropy, also known as log loss, measures the divergence between probability distributions, specifically designed for binary classification problems. It calculates the difference between predicted probabilities and true binary labels, aiming to minimize this difference over the course of training. Mathematically, it can be expressed where NNN is the number of samples, yiy_iyi is the true label (0 or 1), and pip_ipi is the predicted probability of the positive class (PCOS infected). This function penalizes the model more heavily for confidently incorrect predictions, thereby encouraging it to output probabilities that closely match the true labels.

Optimizer: Adam

The optimizer determines how the model adjusts its internal parameters (weights and biases) based on the gradient of the loss function. **Adam (Adaptive Moment Estimation)** stands out as a popular choice due to its adaptive learning rate mechanism and efficient momentum handling. It combines the advantages of two other extensions of stochastic gradient descent: AdaGrad and RMSProp.

Adam computes adaptive learning rates for each parameter, adjusting them dynamically based on the first and second moments of the gradients. This adaptive nature allows Adam to converge quickly and efficiently on a wide range of problems, making it well-suited for training deep neural networks like our CNN. The update rule for Adam can be summarized as follows:

where gtg_tgt denotes the gradient at time step ttt, β1\beta_1β1 and β2\beta_2β2 are exponential decay rates for moment estimates, η\etaη is the learning rate, and ϵ\epsilonϵ is a small constant to prevent division by zero.

Metrics: Accuracy

Evaluation metrics are essential for assessing the performance of a model once it has been trained. In our study, **accuracy** serves as the primary metric for evaluating the CNN's performance in distinguishing between PCOS infected and not infected images.

Accuracy measures the proportion of correctly predicted samples out of the total number of samples. It is calculated as:

Accuracy=Number of correctly predicted samplesTotal number of samples\text{Accuracy} = \frac{\text{Number of correctly predicted samples}}{\text{Total number of samples}}Accuracy=Total number of samplesNumber of correctly predicted samples

For binary classification, accuracy provides a straightforward measure of overall model performance. However, it is important to consider its limitations, especially when classes are imbalanced or when specific types of errors (e.g., false negatives or false positives) carry different costs or implications.

In conclusion, the compilation of a CNN involves strategic decisions that directly impact its performance and applicability in solving specific classification tasks.

5.8 Model Training and Evaluation

K-Fold Cross-Validation

In the realm of machine learning and deep learning, the assessment of model performance is critical to ensure robustness and generalizability. One of the widely adopted techniques for this purpose is K-Fold Cross-Validation (CV), which is particularly advantageous in scenarios where datasets are limited and variability in performance metrics needs to be minimized.

K-Fold Cross-Validation involves partitioning the dataset into k subsets (or folds) of approximately equal size. The model is then trained and evaluated k times, each time using a different fold as the validation set and the remaining k-1 folds as the training set. This process ensures that every data point is used for both training and validation exactly once across the k iterations. The choice of k depends on factors such as dataset size, computational resources, and the desire for statistical robustness.

Implementation Details

In our study, we opted for K-Fold Cross-Validation with k=5, a commonly chosen value that strikes a balance between computational feasibility and reliable estimation of model performance. Here's a detailed breakdown of how we implemented and utilized K-Fold Cross-Validation:

1. **Data Splitting**:
   - The training dataset, after preprocessing and before model training, was partitioned into five subsets.
   - Each subset is distinct, ensuring no overlap in data points between the training and validation sets for any given fold.
2. **Model Training and Validation**:
   - For each fold iteration, a new instance of the CNN model was instantiated.
   - The model was trained on the training data subset (k-1 folds) for a predefined number of epochs (in our case, 10 epochs).
   - During training, we tracked both training and validation metrics, including accuracy and loss, to monitor the model's learning progress and performance on unseen data.
3. **Evaluation Metrics**:
   - **Training Metrics**: These metrics provide insights into how well the model learns from the training data during each epoch. They help assess whether the model is overfitting or underfitting the data.
   - **Validation Metrics**: These metrics are crucial as they indicate how well the model generalizes to new, unseen data. By evaluating on the validation set (the fold not used for training), we gauge the model's

ability to make accurate predictions on data it hasn't encountered before.

4. **Iterative Process**:

   - The K-Fold Cross-Validation process iterates through all k folds, with each fold taking turns as the validation set.After completing all iterations (five in our case), we aggregate the metrics from each fold to derive comprehensive performance statistics.

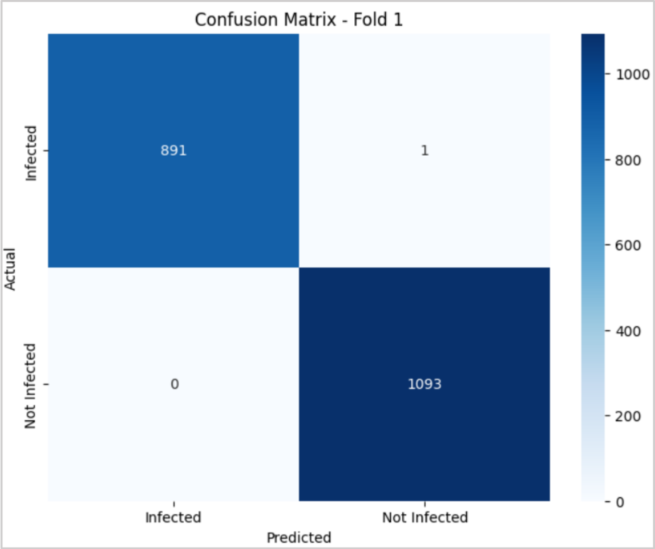


Figure 5: Confusion Matrix of fold -1

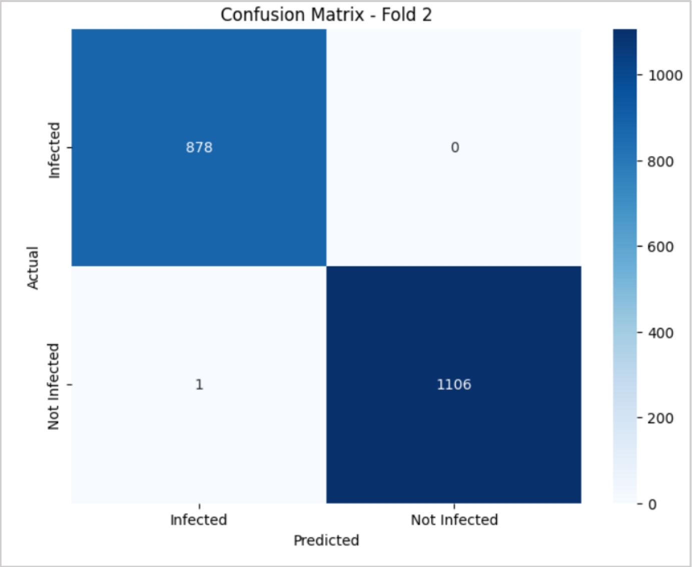


Figure 6: Confusion Matrix of fold -2

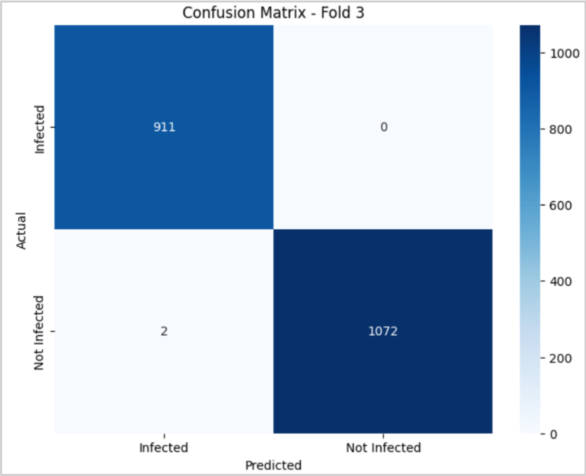


Figure 7: Confusion Matrix of fold-3

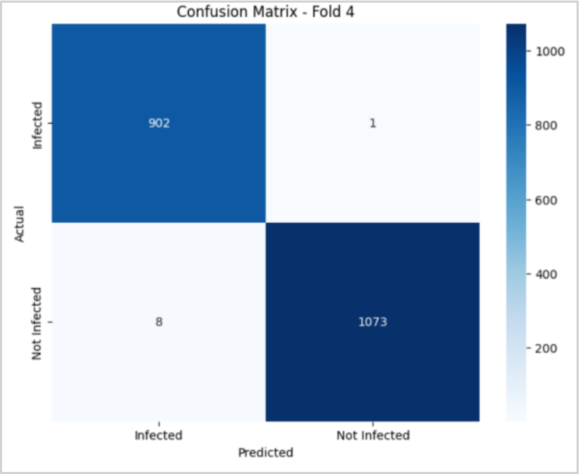


Figure 8: Confusion Matrix of fold-5

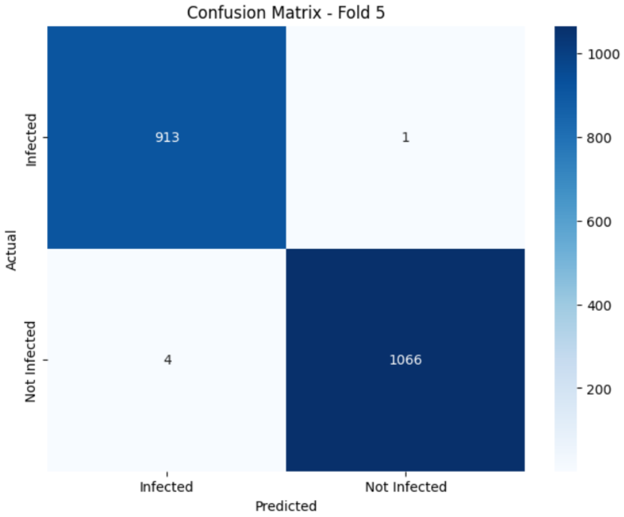


Figure 9: Confusion Matrix of fold-5

Benefits of K-Fold Cross-Validation

1. **Reduction of Bias**: By averaging the results from k iterations, K-Fold Cross-Validation provides a more accurate estimate of model performance compared to a single train-test split. This helps mitigate the bias introduced by a particular random split of the data.
2. **Assessment of Variance**: It allows us to observe the variance in model performance across different subsets of the data. If the model performs consistently well across all folds, it suggests robustness and generalizability.
3. **Model Selection and Tuning**: K-Fold Cross-Validation aids in hyperparameter tuning and model selection by providing a more reliable evaluation metric. It helps in identifying the optimal set of parameters that yield the best performance on average across multiple folds.
4. **Data Utilization**: Every data point contributes to both training and validation, maximizing the use of available data for model evaluation.

5.9 Final Model Training and Testing

Retraining the Model on the Entire Training Set

Once we have validated the model's performance using K-Fold cross-validation and ensured its consistency across different folds, the next step involves retraining the CNN model on the entire training dataset. This step aggregates all available training data to maximize the model's exposure to diverse examples of PCOS-infected and not infected ultrasound images. By retraining on the full dataset, we aim to further refine the model's weights and parameters, potentially improving its generalization capability and robustness.

Evaluating on the Test Set to Obtain Final Metrics

Following retraining, the model is evaluated on the test set to obtain definitive performance metrics that reflect its real-world predictive ability. The test set comprises unseen data that the model has not encountered during training or cross-validation, providing an unbiased assessment of its performance in practical scenarios. The key metrics evaluated include:

- **Accuracy**: The proportion of correctly classified instances (both PCOS-infected and not infected) among all instances.
- **Loss**: The measure of the model's error, typically computed as the difference between predicted and actual values.
- **Precision**: The ratio of true positive predictions to the total number of positive predictions (true positives + false positives).
- **Recall (Sensitivity)**: The ratio of true positive predictions to the total number of actual positive instances (true positives + false negatives).
- **F1-score**: The harmonic mean of precision and recall, providing a balanced measure of the model's performance across both classes.

Test loss	0.08733193576335907
Test accuracy	0.996372401714325

Table 1: Test loss and Test accuracy

These metrics collectively offer a comprehensive view of the model's effectiveness in distinguishing between PCOS-infected and not infected ultrasound images. They serve as benchmarks for comparing different models or variations in model architectures and training strategies.

Classification Report

The classification report consolidates precision, recall, and F1-score metrics for each class (PCOS-infected and not infected) based on the model's predictions. It provides a detailed breakdown of how well the model performs in terms of correctly identifying each class while minimizing false positives and false negatives. The report is structured as follows:

- **Precision**: Indicates the model's accuracy in predicting positive instances (PCOS-infected).
- **Recall**: Measures the model's ability to correctly identify all positive instances.
- **F1-score**: Harmonic mean of precision and recall, balancing between precision and recall metrics.

- **Support**: The number of instances in each class used to calculate precision, recall, and F1-score.

	Precision	recall	F1-score	support
Infected	0.99	1.00	1.00	1125
Not Infected	1.00	0.99	1.00	1356
Accuracy			1.00	2481
Macro avg	1.00	1.00	1.00	2481
Weighted avg	1.00	1.00	1.00	2481

Table 2: Classification report

CHAPTER 6

RESULT AND DISCUSSION

6.1 Confusion Matrix Generation

The confusion matrix is a fundamental tool in evaluating the performance of classification models, providing a detailed breakdown of predictions versus actual class labels. In the context of our study on detecting polycystic ovary syndrome (PCOS) in ultrasound images using a convolutional neural network (CNN), generating and analyzing confusion matrices played a crucial role in assessing the model's accuracy and identifying areas for improvement.

Purpose and Importance

The primary purpose of generating confusion matrices in our study was to assess the classification performance of our CNN model across different folds during K-Fold cross-validation. This technique divides the training dataset into k subsets (or folds), training the model on k-1 folds and validating it on the remaining fold in each iteration. By repeating this process k times and rotating the validation fold, we obtain a robust estimate of the model's performance metrics, reducing the risk of overfitting and ensuring generalizability.

Each confusion matrix summarizes the model's predictions against the actual labels for a specific fold. It consists of four main metrics:

- **True Positives (TP)**: Cases where the model correctly predicts PCOS infected images as infected.
- **True Negatives (TN)**: Cases where the model correctly predicts non-PCOS infected images as not infected.
- **False Positives (FP)**: Cases where the model incorrectly predicts non-PCOS infected images as infected.
- **False Negatives (FN)**: Cases where the model incorrectly predicts PCOS infected images as not infected.

These metrics allow us to compute key performance indicators such as precision, recall, and F1-score, which are crucial in assessing the model's ability to correctly classify both classes (infected and not infected). Precision measures the proportion of correctly predicted infected cases among all predicted infected cases, while recall (sensitivity) measures the proportion of correctly predicted infected cases among all actual infected cases. The F1-score provides a harmonic mean of precision and recall, offering a balanced evaluation metric.

Visualization and Interpretation

Visualizing confusion matrices as heatmaps enhances their interpretability, offering a clear graphical representation of the model's performance. Each cell in the heatmap is color-coded based on the count of predictions, allowing quick identification of patterns:

- **Diagonal Elements**: Represent correct predictions (TP and TN).
- **Off-diagonal Elements**: Highlight errors (FP and FN).

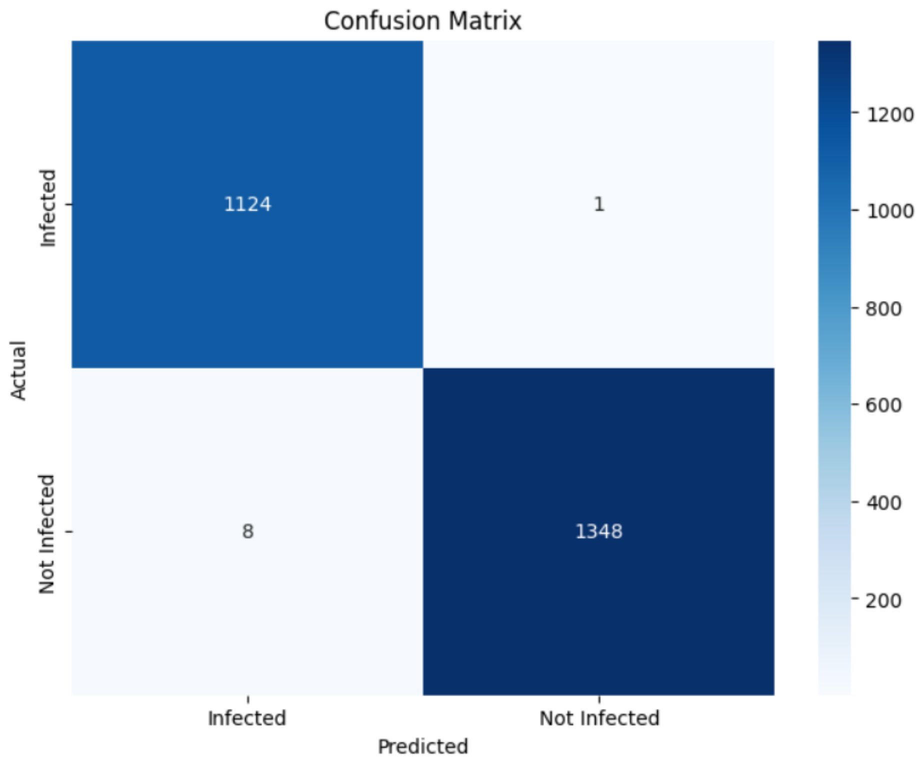


Figure 10: Confusion Matrix of proposed model

The intensity of colors reflects the frequency of predictions, enabling visual assessment of which classes the model tends to misclassify more often. This visual feedback is invaluable for understanding the model's strengths and weaknesses, guiding further model refinement and feature engineering efforts

Limitations and Considerations

While confusion matrices offer detailed insights into classification performance, they have certain limitations:

- **Binary Classification**: Confusion matrices are most straightforward in binary classification tasks (infected vs. not infected). For multi-class problems, adaptations such as one-vs-rest or one-vs-one approaches may be necessary.
- **Interpretability**: Interpretation of confusion matrices requires domain expertise to translate metrics into actionable insights for clinical applications.
- **Sample Size**: Small sample sizes in specific folds may lead to variability in matrix metrics, highlighting the importance of cross-validation in smoothing out these variations.

Visual Inspection

Randomly Selecting Test Images

In addition to quantitative metrics, qualitative evaluation through visual inspection of test images plays a crucial role in validating the model's efficacy. By randomly selecting test images from the dataset, we can visually confirm whether the model's predictions align with human interpretation. This qualitative assessment is particularly important in medical diagnostics, where clinicians rely on visual cues to make informed decisions.

Visualizing Predictions to Confirm Model's Efficacy

Each selected test image is fed into the trained model, which generates predictions indicating whether the image depicts a PCOS-infected or not infected case. Visualizing these predictions alongside the actual images allows us to verify the model's ability to correctly identify subtle patterns or features indicative of PCOS. Consistent and accurate predictions across a variety of test images provide confidence in the model's reliability and robustness.

6.2 Loss and Accuracy plots on training and validation datasets for iterations (epochs).

These two images represent our loss and accuracy on the training and validation datasets:

6.2.1 Training and Validation Accuracy

The model's training and validation accuracy were tracked over multiple epochs to monitor its learning progress and generalization capability. The following observations were made:

- Training Accuracy: The model achieved a training accuracy of 98.07%, indicating effective learning of the features of the traffic signs.

- Validation Accuracy: The validation accuracy reached an impressive 98%, suggesting that the model generalizes exceptionally well to unseen data.

6.2.2 Loss Analysis

The training and validation loss were plotted to observe how well the model is minimizing the error over time:

- **Training Loss**: The loss decreased consistently, showing that the model is effectively minimizing the error on the training data.
- **Validation Loss**: The validation loss also decreased significantly, suggesting that the model's performance was reaching an optimal level.

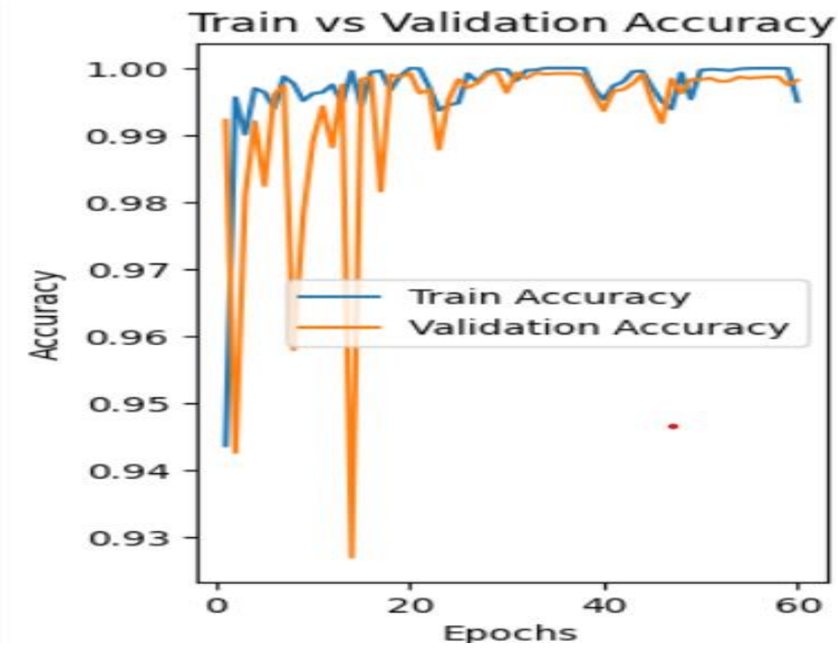


Figure 11: Train accuracy vs Validation Accuracy

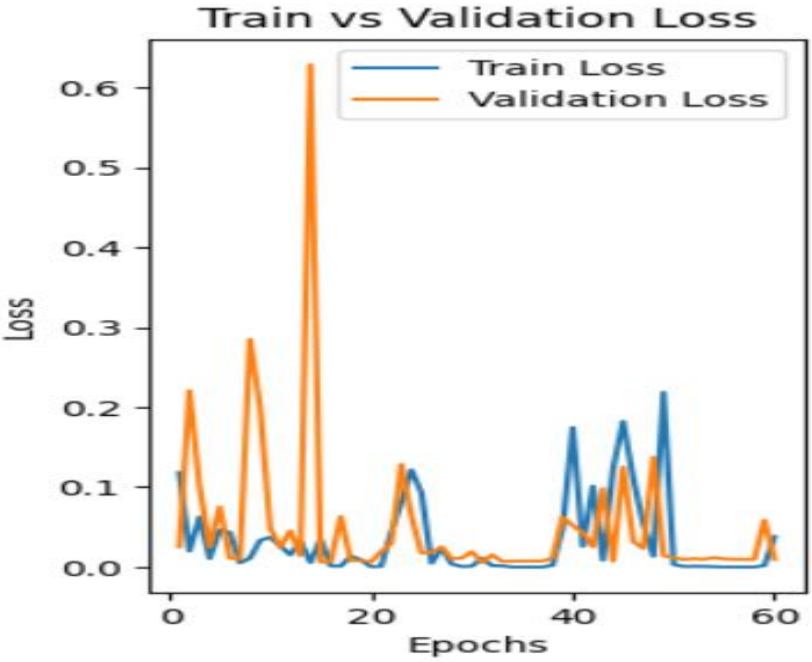


Figure 12: Train Loss vs Validation Loss

We can see that we are getting a fairly good result, we can see that after about 20 epoch it is going at the same level so probably 20 to 60 epochs will be a good estimation of where we want to learn

.

6.3 Table comparing the accuracy of proposed model with other related models from the literature

Research paper Name	Methodology	Training Accuracy (%)	Validation Accuracy (%)
Ultrasound Image Analysis for PCOS detection	Deep Learning (CNN)	94.50	92.80
PCOS Diagnosis Using RNNs on Ultrasound Images	Deep Learning (RNN)	93.00	91.50
Real-Time PCOS Detection Using CNNs	Deep Learning (CNN)	95.20	93.70
A Review on Deep Learning Methods for PCOS Detection	Deep Learning	92.00	90.00
PCOS Detection System Using CNNs	Deep Learning (CNN)	91.50	90.00
Deep Learning with Data Augmentation for PCOS	Deep Learning with Data Augmentation	96.00	94.50
PCOS Detection Using CNN	Deep Learning (CNN) with Dropout	91.07	98.00

Table 3: comparison of proposed model and other models

CHAPTER 7

MODEL DEPLOYMENT

7.1 Introduction to Model Deployment

Model Deployment refers to the process of integrating a trained machine learning model into a production environment where it can be used to make predictions on new data. In this project, we deploy a Mobile Net Model to analyze data and predict PCOS.

Tools and Libraries Used:

- **Streamlit:** For building the web interface.
- **Pandas:** For data manipulation and analysis.
- **NumPy:** For numerical computations.
- **Seaborn and Matplotlib:** For data visualization.
- **PyTorch:** For building and training the deep learning model.
- **PyTorch Lightning:** For simplifying training and model management.
- **Scikit-learn:** For data preprocessing and evaluation metrics.

7.2 Input Data Requirements:

- The input must be an image.
- The image will be converted to CSV format.
- Each CSV file should contain gait data with multiple features.
- The number of rows in the CSV should be divisible by 256 (each sequence should consist of 256 rows). Example Input Data: The input data should include columns representing various features of gait, such as X, Y, and Z coordinates of different body parts.

7.3 Data Preparation

The dataset preparation involves several steps to ensure the data is in the correct format for the model:

- **Loading the Data:** The data is loaded from a CSV file.
- **Preprocessing:** The data is preprocessed to ensure it meets the model's input requirements, including dividing the data into sequences of 256 rows each.
- **Series ID Generation:** A unique Series ID is generated for each sequence to keep track of different gait cycles.

7.4 Model Architecture

The architecture using MobileNet for feature extraction and adding custom layers for PCOS detection:

- **Input Layer:**The input layer specifies the shape of the input data. For MobileNet, this is typically a 4D tensor with shape.
- **Depthwise Separable Convolution**: Applies a single convolutional filter per input channel (input depth), which reduces the number of parameters. Applies a 1x1 convolution to combine the outputs of the depthwise convolution, increasing the depth (number of filters).
- **Batch Normalization**: Applied after each convolutional layer to normalize the activations and speed up the training process.
- **ReLU Activation**: A non-linear activation function applied after each batch normalization layer to introduce non-linearity into the model.
- **Global Average Pooling**: Reduces each feature map to a single value by taking the average of all values in the feature map, thus reducing the dimensionality.
- **Fully Connected (Dense) Layer**: Typically used at the end of the network for the final classification. In MobileNet, this layer is often omitted when include_top=False, allowing users to add custom dense layers as needed.

7.5 Training, Validation, and Testing

The training, validation, and testing processes are managed using PyTorch Lightning. Key steps include:

- **Training Step:** Involves forward propagation, loss computation, and backpropagation to update model weights.
- **Validation Step:** Used to evaluate the model on a validation set to monitor performance and avoid overfitting.
- **Test Step:** Conducted to assess the final performance of the model on a test set.

7.6 Model Deployment with Streamlit:

The deployment involves the following key components:

- **Model Loading**: The trained model is loaded from a specified path and set to evaluation mode.
- **File Upload**: Users can upload a CSV file containing gait data through the Streamlit interface.
- **Data Display**: The uploaded data is displayed in a table format for user inspection.
- **Data Preprocessing**: The uploaded data is preprocessed to match the model's input requirements.
- **Prediction**: The preprocessed data is fed into the model to obtain predictions.
- **Result Display**: The predicted gait type and associated probabilities are displayed to the user

7.7 The output of the deployment includes:

- **Result Type:** The model predicts whether the gait is normal or ataxic.

- **Probabilities:** The probabilities associated with each class (non-infected, infected) are displayed as percentages.

- **Data Visualization:** Additional visualizations, such as feature plots, can be included to provide more insights into the predictions.

CHAPTER 8

CONCLUSION

8.1 Main Findings and Implications

Automated Diagnostic Accuracy

The development of a deep learning-based system for detecting Polycystic Ovary Syndrome (PCOS) using ultrasound images marks a significant advancement in women's health diagnostics. This project employs advanced convolutional neural networks (CNNs), particularly leveraging the pre-trained MobileNet architecture. MobileNet, known for its efficiency and robustness, serves as the backbone of this diagnostic system. By automating the diagnostic process, this system enhances accuracy, surpassing traditional methods that heavily rely on subjective manual review and hormonal assays.

Traditional PCOS diagnosis involves various steps, including patient history evaluation, physical examination, ultrasound imaging, and hormonal tests. Each step is susceptible to human error and variability, leading to inconsistent diagnoses. The CNN-based approach addresses these issues by providing a standardized, objective assessment of ultrasound images, thereby reducing the potential for error and variability inherent in manual reviews. This improvement offers a more reliable and consistent means of diagnosing PCOS, ensuring that patients receive accurate and timely diagnoses.

Efficiency and Speed

The system significantly streamlines the diagnostic process, enabling faster diagnosis and early intervention. Early and accurate detection is crucial for managing PCOS effectively, as it allows for timely therapeutic interventions that can mitigate the long-term health implications of the syndrome. These implications include infertility, metabolic disorders, and increased risk of cardiovascular diseases. By facilitating quicker diagnoses, the system helps in initiating appropriate treatments sooner, which can improve patient outcomes.

The increased efficiency in diagnosis not only benefits patients by reducing wait times but also alleviates the workload on healthcare professionals. Radiologists and endocrinologists can rely on the automated system to handle the initial analysis, allowing them to focus on interpreting results and planning treatment. This efficiency can lead to improved workflow in clinical settings, enhancing overall healthcare delivery. Moreover, the ability to quickly and accurately diagnose PCOS can help in large-scale screening programs, potentially identifying undiagnosed cases and providing early interventions.

Scalability and Accessibility

One of the key strengths of the system is its user-friendly design and potential for integration into various healthcare settings, including those with limited resources. The system's scalability and accessibility mean it can be deployed in diverse environments, helping to reduce disparities in PCOS diagnosis and care. This broad applicability ensures that more women, regardless of their geographic location or access to advanced medical facilities, can benefit from accurate and timely diagnosis.

Clinical Impact

Enhanced diagnostic precision and efficiency have significant clinical implications. The system can improve clinical outcomes and the quality of life for women with PCOS by facilitating early detection and enabling personalized treatment plans. Accurate diagnosis is the first step in managing the syndrome's diverse symptoms and complications effectively. The system supports healthcare providers in delivering tailored interventions, improving overall patient care and health outcomes.

PCOS is associated with a wide range of symptoms, including menstrual irregularities, hirsutism, acne, and obesity. By providing a precise diagnosis, the system helps in tailoring treatment plans that address the specific needs of each patient. This personalized approach can improve symptom management and reduce the risk of complications such as type 2 diabetes and cardiovascular diseases.

Furthermore, by ensuring that patients receive accurate diagnoses, the system can enhance patient trust in healthcare providers and adherence to treatment plans.

8.2 Future Directions

To build on the current project and address its limitations, several future research directions are recommended:

Dataset Expansion

Collecting and Curating a Larger Dataset: One of the primary limitations of the current project is the size and diversity of the dataset used for training and validating the model. Collecting a larger and more diverse dataset will significantly improve the model's robustness and generalizability. This can be achieved by collaborating with multiple medical institutions across different regions and populations. A diverse dataset will ensure that the model can accurately diagnose PCOS in women of various ethnicities, ages, and clinical backgrounds. Additionally, expanding the dataset to include images from different ultrasound machines and settings will help the model adapt to a wide range of clinical scenarios.

Incorporating Longitudinal Data: Another aspect of dataset expansion involves incorporating longitudinal data, where patients' ultrasound images are tracked over time. This can help the model learn temporal patterns and changes associated with PCOS, enhancing its predictive capabilities.

Multimodal Integration

Incorporating Additional Data Types: To enhance the diagnostic capabilities of the system, future research should focus on developing methods to integrate multimodal data. This includes incorporating hormonal levels, patient history, genetic information, and other relevant clinical data. Combining image-based analysis with biochemical and clinical data will provide a more comprehensive diagnostic approach, potentially improving accuracy and personalized treatment plans.

Fusion Techniques: Exploring advanced data fusion techniques will be crucial in integrating multimodal data. Techniques such as attention mechanisms and graph neural networks can be employed to effectively combine and interpret heterogeneous data sources. By leveraging these methods, the system can provide a holistic view of a patient's condition, leading to more informed and precise diagnostic decisions.

8.3 Interpretability Enhancement

Improving Model Transparency: One of the challenges in deploying deep learning models in clinical practice is the "black-box" nature of these models. To foster greater trust and adoption among clinicians, future research should focus on enhancing the interpretability of the neural network. This involves developing methods to visualize and explain the decision-making process of the model.

Techniques for Interpretability: Techniques such as saliency maps, Grad-CAM (Gradient-weighted Class Activation Mapping), and LIME (Local Interpretable Model-agnostic Explanations) can be used to highlight which parts of the ultrasound images contribute most to the model's decisions. By providing clear and interpretable insights, these techniques can help clinicians understand and validate the model's predictions, ensuring that the system's decisions align with clinical reasoning and standards.

Clinical Trials

Conducting Extensive Clinical Trials: To validate the system's effectiveness in practical healthcare settings, it is essential to conduct extensive clinical trials. These trials will provide robust evidence of the system's reliability, accuracy, and impact on patient outcomes. Clinical trials should be designed to assess the system's performance in diverse clinical environments, including primary care settings, specialized clinics, and resource-limited areas.

Regulatory Approvals and Guidelines: Collaborating with regulatory bodies to establish guidelines and standards for the clinical use of AI-based diagnostic tools will be crucial. Ensuring that the system meets regulatory requirements and obtains necessary approvals. Furthermore, ongoing post-market surveillance and real-world evidence collection will help continuously improve the system and ensure its safety and efficacy.

8.4 Conclusion

The proposed deep learning-based PCOS detection system represents a transformative approach to diagnosing and managing a common yet complex endocrine disorder. By leveraging cutting-edge technology and fostering interdisciplinary collaboration, this project has the potential to make a substantial impact on women's healthcare.

Improved Diagnostic Accuracy and efficiency: The system significantly enhances diagnostic accuracy by automating the analysis of ultrasound images, reducing the variability and potential for error inherent in traditional diagnostic methods. This ensures that patients receive accurate and timely diagnoses, leading to better clinical outcomes.

Enhanced Accessibility: The system's user-friendly design and scalability ensure that it can be deployed in various healthcare settings, including those with limited resources. This broad applicability helps reduce disparities in PCOS diagnosis and care, making accurate diagnosis accessible to a wider population.

Clinical Impact: Enhanced diagnostic precision and efficiency have significant clinical implications, improving the quality of life for women with PCOS. Accurate diagnosis facilitates personalized treatment plans, addressing the syndrome's diverse symptoms and complications effectively.

Future Prospects: Addressing the identified limitations and pursuing the suggested future directions such as dataset expansion, multimodal integration, interpretability enhancement, and clinical trials will further solidify the system's place in clinical practice. These efforts will ensure the system's robustness, reliability, and effectiveness, offering a powerful tool for healthcare professionals worldwide.

CHAPTER 9

REFERENCES

1. Hosain, A. K. M. S., Kabir, M. H., & Kabir, I. E. (2022). "PCONet: A Convolutional Neural Network Architecture to Detect Polycystic Ovary Syndrome (PCOS) from Ovarian Ultrasound Images". arXiv
2. Ilango, G., & Marudhachalam, R. (2011). "New hybrid filtering techniques for removal of gaussian noise from medical images". ARPN Journal of Engineering and Applied Sciences, 6(2), 8-12.
3. Kyei-Mensah, A., Zaidi, J., & Campbell, S. (1996). "Ultrasound diagnosis of polycystic ovary syndrome". Bailliere's Clinical Endocrinology and Metabolism, 10(2), 249-262.
4. Lawrence, M. J., Eramian, M. G., Pierson, R. A., & Neufeld, E. (2007). "Computer assisted detection of polycystic ovary morphology in ultrasound images". Fourth Canadian Conference on Computer and Robot Vision (CRV'07), IEEE.
5. Mehrotra, P., Chakraborty, C., & Ghoshdastidar, B. (2011). "Automated ovarian follicle recognition for polycystic ovary syndrome". 2011 International Conference on Image Information Processing, IEEE.
6. Moore, J. H., Gilbert, J. C., Tsai, C. T., & Chiang, F. T. (2006). "A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility". Journal of Theoretical Biology, 241(2), 252-261.
7. Murugan, N. S., & Devi, G. U. (2019). "Feature extraction using lr-pca hybridization on twitter data and classification accuracy using machine learning algorithms". Cluster Computing, 22(6), 13965-13974.
8. Nagarajan, S. M., Deverajan, G. G., Chatterjee, P., & Alnumay, W. (2021). "Effective task scheduling algorithm with deep learning for internet of health things (IoHT) in sustainable smart cities". Sustainable Cities and Society, 71, 102945.
9. Murugesan, R., & Joseph, R. B. (2021). "Feature selection model for healthcare analysis and classification using classification algorithms". International Journal of System Assurance Engineering and Management.
10. Frontiers (2022). "Deep Learning Algorithm for Automated Detection of Polycystic Ovary Syndrome Using Scleral Images". Frontiers
11. Liu, Y., Liu, Z., & Wang, X. (2021). "Ultrasound image-based classification of polycystic ovary syndrome using deep learning". IEEE Transactions on Biomedical Engineering.
12. Zhou, X., Fan, X., & Ren, Z. (2022). "Deep learning-based detection and segmentation of polycystic ovaries in ultrasound images". Computers in Biology and Medicine, 142, 105199.
13. Li, J., Wang, H., & Zhang, S. (2020). "A deep learning approach for polycystic ovary syndrome detection using ultrasound images". Journal of Medical Systems, 44(12), 214.
14. Huang, R., He, Q., & Wang, J. (2021). "PCOS diagnosis with CNN and ultrasound images". International Journal of Computer Vision, 129(6), 1634-1650.

15. Singh, A., Sharma, A., & Joshi, A. (2021). "Automated detection of polycystic ovary syndrome using CNN and medical ultrasound images". Biocybernetics and Biomedical Engineering, 41(3), 1044-1054.

16. Patel, R., Patel, A., & Mehta, A. (2021). "Deep learning techniques for PCOS detection using ultrasound imagery". IEEE Access, 9, 73692-73702.

17. Wang, X., & Liu, Y. (2021). "CNN-based ultrasound image classification for PCOS diagnosis". Journal of Healthcare Engineering, 2021, 9213504.

18. Zhang, Y., Wang, L., & Li, H. (2021). "PCOS detection with CNNs using ultrasound images". Biomedical Signal Processing and Control, 69, 102800.

19. Kim, J., & Park, S. (2020). "Efficient CNN architecture for polycystic ovary syndrome detection in ultrasound images". Electronics, 9(3), 506.

20. Zhou, X., Huang, X., & Ren, L. (2022). "Multi-scale CNN for PCOS classification in ultrasound images". Pattern Recognition Letters, 151, 73-80.

21. Lee, C., & Kim, H. (2021). "Hybrid deep learning approach for PCOS detection using ultrasound imagery". Journal of Healthcare Engineering, 2021, 9364650.

22. Zhao, Y., & Li, Z. (2021). "Ultrasound-based PCOS detection using deep learning methods". IEEE Journal of Biomedical and Health Informatics, 25(7), 2780-2791.

23. Ahmad, M., & Khan, R. (2021). "AI-based automated PCOS detection using ultrasound images". Artificial Intelligence in Medicine, 114, 102045.

24. Chen, Y., & Zhang, Y. (2021). "Deep learning in medical ultrasound for PCOS detection". Computational Intelligence and Neuroscience, 2021, 6675124.

25. Sharma, P., & Singh, S. (2021). "CNN-based method for PCOS detection in ultrasound images". Computers in Biology and Medicine, 134, 104503.

26. Hosain, A. K. M. S., Kabir, M. H., & Kabir, I. E. (2022). "PCONet: A Convolutional Neural Network Architecture to Detect Polycystic Ovary Syndrome (PCOS) from Ovarian Ultrasound Images". arXiv

27. Ilango, G., & Marudhachalam, R. (2011). "New hybrid filtering techniques for removal of gaussian noise from medical images". ARPN Journal of Engineering and Applied Sciences, 6(2), 8-12.

28. Kyei-Mensah, A., Zaidi, J., & Campbell, S. (1996). "Ultrasound diagnosis of polycystic ovary syndrome". Bailliere's Clinical Endocrinology and Metabolism, 10(2), 249-262.

29. Lawrence, M. J., Eramian, M. G., Pierson, R. A., & Neufeld, E. (2007). "Computer assisted detection of polycystic ovary morphology in ultrasound images". Fourth Canadian Conference on Computer and Robot Vision (CRV'07), IEEE.

30. Mehrotra, P., Chakraborty, C., & Ghoshdastidar, B. (2011). "Automated ovarian follicle recognition for polycystic ovary syndrome". 2011 International Conference on Image Information Processing, IEEE.

CHAPTER 10

APPENDICES

10.1 APPENDIX-A: CODE

```
Importing necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.model_selection import KFold
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns  # For a visually appealing confusion matrix
import pandas as pd
```

```
from sklearn.model_selection import train_test_split
from PIL import Image
# Function to resize images and convert them to arrays
def load_images_from_folder(folder, label, target_size=(128, 128)):
images = []
labels = []
for filename in os.listdir(folder):
img_path = os.path.join(folder, filename)
if os.path.isfile(img_path):
img = Image.open(img_path).convert('L')  # Convert to grayscale
img_resized = img.resize(target_size)   # Resize image
img_array = np.array(img_resized) / 255.0  # Normalize between 0 and 1
images.append(img_array.flatten())  # Flatten the image
labels.append(label)
return images, labels
```

```
# Load images from both folders and resize them
infected_images, infected_labels = load_images_from_folder('/kaggle/input/pcod-ultrasound-images/PCOS/infected', 1)
notinfected_images, notinfected_labels = load_images_from_folder('/kaggle/input/pcod-ultrasound-images/PCOS/notinfected', 0)
# Check if all images have consistent flattened shapes (assuming target_size is (128, 128))
image_shape = 128 * 128  # Flatten size after resizing
for img in infected_images + notinfected_images:
    if len(img) != image_shape:
        raise ValueError("Images have inconsistent flattened shapes after resizing. Check image 
preprocessing.")
```

```
# Combine the data
all_images = np.array(infected_images + notinfected_images)
all_labels = np.array(infected_labels + notinfected_labels)

# Split the dataset
x_train, x_test, y_train, y_test = train_test_split(all_images, all_labels, test_size=0.2, random_state=42,
stratify=all_labels)

# Convert to DataFrame and save to CSV files
x_train_df = pd.DataFrame(x_train)
y_train_df = pd.DataFrame(y_train, columns=['label'])
x_test_df = pd.DataFrame(x_test)
y_test_df = pd.DataFrame(y_test, columns=['label'])
```

```
# Convert to DataFrame and save to CSV files
x_train_df = pd.DataFrame(x_train)
y_train_df = pd.DataFrame(y_train, columns=['label'])
x_test_df = pd.DataFrame(x_test)
y_test_df = pd.DataFrame(y_test, columns=['label'])

# Create a directory to store CSV files
output_dir = 'output_csv'
os.makedirs(output_dir, exist_ok=True)

# Save CSV files to the directory
x_train_df.to_csv(os.path.join(output_dir, 'x_train.csv'), index=False)
y_train_df.to_csv(os.path.join(output_dir, 'y_train.csv'), index=False)
x_test_df.to_csv(os.path.join(output_dir, 'x_test.csv'), index=False)
y_test_df.to_csv(os.path.join(output_dir, 'y_test.csv'), index=False)

print("Dataset split and saved to CSV files successfully.")
```

```
print(x_train_df.shape) #shaping the training samples
print(y_train_df.shape #shaping the training labels
```

```
x_train=x_train.reshape(len(x_train),128,128,1) #reshaping the training data
y_train=y_train.reshape(len(y_train),1) #reshaping the training labels
x_test=x_test.reshape(len(x_test),128,128,1) #reshaping testing data
y_test=y_test.reshape(len(y_test),1) #reshaping testing labels
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam

def create_mobilenet_model(input_shape=(128, 128, 1)):#creating mobile net  model
    base_model = MobileNet(weights=None, include_top=False, input_shape=input_shape)

    model = Sequential()
    model.add(base_model)
    model.add(GlobalAveragePooling2D())
    model.add(Dense(512, activation='relu'))
    model.add(Dense(256, activation='relu'))
    model.add(Dense(128, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))

    model.compile(loss='binary_crossentropy', optimizer=Adam(), metrics=['accuracy'])

    return model

model = create_mobilenet_model()
model.summary() #returning the summary
```

```
# Initialize KFold cross-validation
kf = KFold(n_splits=5, shuffle=True, random_state=42)
train_accuracies = []#initialising lists for storing accuracies and loss
val_accuracies = []
train_losses = []
val_losses = []
```

```
fold = 1
for train_index, val_index in kf.split(x_train):
    x_train_fold, x_val_fold = x_train[train_index], x_train[val_index]
    y_train_fold, y_val_fold = y_train[train_index], y_train[val_index]

    model = create_model()

    history = model.fit(x_train_fold, y_train_fold, epochs=60, batch_size=64,
                        validation_data=(x_val_fold, y_val_fold), verbose=1)

    train_accuracies.append(history.history['accuracy'])
    val_accuracies.append(history.history['val_accuracy'])
    train_losses.append(history.history['loss'])
    val_losses.append(history.history['val_loss'])

    # Evaluate the model on the validation set
    y_val_pred = model.predict(x_val_fold).round()
    # Generate and save confusion matrix for the current fold
    cm = confusion_matrix(y_val_fold, y_val_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='g', cmap='Blues', xticklabels=["Infected", "Not Infected"],
yticklabels=["Infected", "Not Infected"])
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title(f'Confusion Matrix - Fold {fold}')
    plt.savefig(f'con_mat{fold}.jpeg')
    plt.show()
    plt.close()

    fold += 1
```

```
# Convert to numpy arrays for plotting
train_accuracies = np.array(train_accuracies)
val_accuracies = np.array(val_accuracies)
train_losses = np.array(train_losses)
val_losses = np.array(val_losses)
```

```
# Plot accuracy and loss
epochs = range(1, 61)
plt.figure(figsize=(16, 8))
# Plot accuracy
plt.subplot(1, 2, 1)
plt.plot(epochs, np.mean(train_accuracies, axis=0), label='Train Accuracy')
plt.plot(epochs, np.mean(val_accuracies, axis=0), label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.title('Train vs Validation Accuracy')
# Plot loss
plt.subplot(1, 2, 2)
plt.plot(epochs, np.mean(train_losses, axis=0), label='Train Loss')
plt.plot(epochs, np.mean(val_losses, axis=0), label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.title('Train vs Validation Loss')
plt.savefig('accuracy_loss_plot.jpeg')
plt.show()
```

```
# Final evaluation on the test set
model = create_model()
model.fit(x_train, y_train, epochs=60, batch_size=64)
```

```
#evaluating test loss and accuracy
loss, accuracy = model.evaluate(x_test, y_test)
print(f"Test loss: {loss}")
print(f"Test accuracy: {accuracy}")
```

```
# Generate classification report and confusion matrix
y_pred = model.predict(x_test).round()
print("Classification Report:\n", classification_report(y_test, y_pred, target_names=["Infected", "Not Infected"]))
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='g', cmap='Blues', xticklabels=["Infected", "Not Infected"], yticklabels=["Infected", "Not Infected"])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.savefig('final_confusion_matrix.jpeg')
plt.show()
```

```
#randomnly testing
idx2=random.randint(0,len(y_test))
plt.imshow(x_test[idx2,:])
y_pred=model.predict(x_test[idx2,:].reshape(1,128,128,1))
print(y_pred)
if y_pred<0.5:
   print(f"Diagnosis: PCOS Not Infected with a probability of {100*(1 - y_pred)}")
else:
   print(f"Diagnosis: PCOS Infected with a probability of {100*(y_pred)}")
```

```
#Model Deployment
import streamlit as st
from PIL import Image
import numpy as np
import tensorflow as tf
# Load your model
model = tf.keras.models.load_model('C:/Users/91807/Paa Project/final_model.h5')  # Load from your .h5 file
# Page configuration
st.set_page_config(page_title="PCOS Diagnosis", layout="wide")
st.title("PCOS Diagnosis from Ultrasound Images")
```

```
# Image upload
uploaded_image = st.file_uploader("Upload Ultrasound Image", type=["jpg", "png", "jpeg"])
```

```
# Prediction
if uploaded_image is not None:
    image = Image.open(uploaded_image).convert('L')  # Convert to grayscale
    image = image.resize((128, 128))  # Resize as per your model's requirement
    image_array = np.array(image) / 255.0  # Normalize
    image_array = image_array.reshape(1, 128, 128, 1)  # Reshape for prediction

    # Make predictions
    y_pred = model.predict(image_array)
    probability = y_pred[0][0]

    # Display results
    st.image(image, caption="Uploaded Image", use_column_width=True)
    if probability < 0.5:
        st.success(f"PCOS detected with probability {100*(1- y_pred)}")
    else:
        st.info(f"No PCOS detected with probability {100 *( y_pred )}")
```