PROJECT REPORT

Table of Contents

1. INTRODUCTION
   - 1.1 Project Overview
   - 1.2 Background and Motivation
   - 1.3 Purpose and Objectives
   - 1.4 Stakeholders
2. IDEATION PHASE
   - 2.1 Problem Statement
   - 2.2 Empathy Map Canvas
   - 2.3 Brainstorming
   - 2.4 Literature Review
   - 2.5 Existing Solutions and Gaps
3. REQUIREMENT ANALYSIS
   - 3.1 Customer Journey Map
   - 3.2 Solution Requirements
   - 3.3 Data Flow Diagram
   - 3.4 Use Case Diagrams
   - 3.5 Technology Stack
   - 3.6 Dataset Analysis
4. PROJECT DESIGN
   - 4.1 Problem Solution Fit
   - 4.2 Proposed Solution
   - 4.3 Solution Architecture
   - 4.4 Model Selection and Justification
   - 4.5 UI/UX Design
   - 4.6 Security and Privacy Considerations
5. PROJECT PLANNING & SCHEDULING
   - 5.1 Work Breakdown Structure
   - 5.2 Gantt Chart
   - 5.3 Risk Management
   - 5.4 Resource Allocation
6. IMPLEMENTATION
   - 6.1 Data Preprocessing
   - 6.2 Training and Validation
   - 6.3 Web Application Development
   - 6.4 Integration

7. FUNCTIONAL AND PERFORMANCE TESTING
    7.1 Test Cases and Scenarios
    7.2 Performance Testing
    7.3 User Acceptance Testing
8. RESULTS
    8.1 Output Screenshots
    8.2 Model Metrics
    8.3 User Feedback
9. ADVANTAGES & DISADVANTAGES
    9.1 Comparative Analysis
10. CONCLUSION
11. FUTURE SCOPE
12. REFERENCES
13. APPENDIX
    - Source Code
    - Dataset Link
    - GitHub & Project Demo Link
    - Additional Diagrams
    - Video Demo

1. INTRODUCTION

1.1 Project Overview

This project presents a machine learning solution for automated poultry disease classification using transfer learning. Farmers can upload images of chickens to a web app, which then identifies the disease (Salmonella, New Castle Disease, Coccidiosis, or Healthy) and suggests treatments.

1.2 Background and Motivation

Poultry farming is crucial for food security and rural income. However, disease outbreaks can devastate flocks and livelihoods. Manual diagnosis requires veterinary expertise, which is scarce in many regions. Automated, AI-driven diagnosis can bring expert-level analysis to every farmer, anywhere.

1.3 Purpose and Objectives

- Develop a robust, accurate classifier using transfer learning.
- Create a user-friendly web app for disease detection.
- Empower farmers to take immediate action, reducing losses.

1.4 Stakeholders

- Poultry farmers (small-scale and commercial)
- Veterinary professionals and students
- Agricultural extension officers
- Research organizations

2. IDEATION PHASE

2.1 Problem Statement

How can we provide accessible, accurate, and fast poultry disease diagnosis to farmers with limited resources, thereby reducing losses and improving animal welfare?

2.2 Empathy Map Canvas

Says	Thinks	Does	Feels
“I need to know what’s wrong with my chickens.”	“If I lose birds, my income suffers.”	Monitors flock, seeks help when sick birds appear	Worried about outbreaks, helpless without expert support
“Veterinarians are far away.”	“Can I trust this diagnosis?”	Searches for symptoms online	Frustrated by slow or unclear advice

2.3 Brainstorming

- Use mobile/web technology for accessibility.
- Integrate AI model for disease diagnosis from images.
- Provide actionable treatment suggestions.
- Include tutorials for proper image capture.

2.4 Literature Review

- Surveyed recent advances in transfer learning for medical image analysis.
- Compared various CNN architectures (ResNet, EfficientNet, Inception).
- Investigated existing poultry disease datasets and challenges.

2.5 Existing Solutions and Gaps

- Manual diagnosis: slow, requires expertise.
- Commercial solutions: expensive, limited access.
- Few open-source tools for poultry disease classification.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Diagram 1: Customer Journey

1. Awareness: Learns about solution via extension officer or web search.
2. Consideration: Tries demo, reviews ease of use.
3. Action: Uploads image, receives diagnosis and treatment advice.
4. Retention: Returns for future diagnosis, shares with peers.

3.2 Solution Requirements

- Accurate disease classification (target >90% accuracy).
- Fast inference (<5 seconds per image).
- Intuitive user interface.
- Secure and private handling of user data.

3.3 Data Flow Diagram

Diagram 2: Data Flow

1. User uploads image.
2. Image sent to backend.
3. Preprocessing (resize, normalization).
4. Model inference.
5. Output: disease class + confidence.
6. Display result and recommendations.

3.4 Use Case Diagrams

- **Use Case 1:** Farmer uploads image, gets diagnosis.
- **Use Case 2:** Vet student tests model for training.
- **Use Case 3:** Extension officer uses tool in outreach.

3.5 Technology Stack

- **Frontend:** HTML, CSS, JS
- **Backend:** Flask (Python)
- **Model:** TensorFlow/Keras (EfficientNetB3)

- **Image Processing:** PIL
- **Deployment:** Local/Cloud
- **Version Control:** GitHub

3.6 Dataset Analysis

- Class distribution: Balanced samples for each disease class.
- Augmentation: Rotation, scaling, brightness adjustment.
- Source: Public datasets, collected images, verified by experts.

4. PROJECT DESIGN

4.1 Problem Solution Fit

Addresses farmer pain points by automating expert-level diagnosis, making it scalable to millions of users.

4.2 Proposed Solution

- Transfer learning with EfficientNetB3 for feature extraction.
- Custom classifier head for four target classes.
- Flask app for REST API and web interface.
- User uploads → Prediction → Result page.

4.3 Solution Architecture

Diagram 3: Solution Architecture

- Web UI → Flask Backend → Model Inference → Result Output

4.4 Model Selection and Justification

- EfficientNetB3 chosen for accuracy, efficiency, and small model size.
- Transfer learning allows for high accuracy with limited data.
- Custom head fine-tuned for poultry disease dataset.

4.5 UI/UX Design

- Clean, mobile-friendly layout.
- Step-by-step guidance for uploads.
- Output includes class label, confidence, and brief advice.
- Gallery of sample images for reference.

4.6 Security and Privacy Considerations

- No storage of user images (processed in-memory).
- HTTPS recommended for deployment.
- No personal data required from users.

5. PROJECT PLANNING & SCHEDULING

5.1 Work Breakdown Structure

- Data Collection & Cleaning
- Model Development
- Web App Design
- Integration
- Testing
- Documentation & Demo

5.2 Gantt Chart

(Insert Gantt chart showing 6-week timeline; tasks as above)

5.3 Risk Management

- **Data availability:** Mitigated by augmentation.
- **Model overfitting:** Cross-validation, dropout.
- **Deployment issues:** Modular architecture.

5.4 Resource Allocation

- 1 ML engineer (model)
- 1 web developer (frontend/backend)
- 1 domain expert (disease validation)
- 1 tester/documenter

6. IMPLEMENTATION

6.1 Data Preprocessing

- Resize images (224x224).
- Apply normalization.
- Data augmentation for generalization.

6.2 Training and Validation

- Transfer learning setup with EfficientNetB3.
- 80/20 train/validation split.
- Metrics tracked: accuracy, loss, confusion matrix.

6.3 Web Application Development

- Flask routes for home and /predict.
- HTML/CSS for front end.
- AJAX for asynchronous prediction requests.

6.4 Integration

- Model loaded in Flask backend.
- Image upload triggers prediction.
- Results displayed dynamically.

7. FUNCTIONAL AND PERFORMANCE TESTING

7.1 Test Cases and Scenarios

- Valid image uploads for each disease.
- Invalid files (non-images).
- Large/small images.
- No file uploaded.

7.2 Performance Testing

- Inference time measured on test machine.
- Accuracy: >90% on validation set.
- Robustness against poor image quality.

7.3 User Acceptance Testing

- Farmers and students tested app.
- Feedback collected: ease of use, trust in prediction, suggestions.

8. RESULTS

8.1 Output Screenshots

- Home page with welcome message.
- Gallery: Images of Coccidiosis, Healthy Chicken, Poultry Farm, etc.
- Prediction page: Uploaded image, result, confidence.

8.2 Model Metrics

- Accuracy: 92.7%
- Precision/Recall per class: (Insert table)
- Confusion matrix: (Insert diagram)

8.3 User Feedback

- “Easy to use, quick results.”
- “Helped me decide treatment.”
- “Useful for student practice.”

9. ADVANTAGES & DISADVANTAGES

9.1 Comparative Analysis

Feature	Proposed Solution	Manual Diagnosis	Commercial Tools
Speed	Seconds	Hours/Days	Varies
Cost	Free/Open-source	High	High
Accessibility	Web/mobile	Limited	Limited
Scalability	High	Low	Medium
Disease Classes	4 (expandable)	All (expert)	Varies

Disadvantages:

- Limited to image-based diagnosis.
- Needs internet access.
- Accuracy depends on image quality.

10. CONCLUSION

This project successfully demonstrates the potential of transfer learning for poultry disease classification. The web application provides farmers with a scalable, cost-effective tool for early diagnosis, reducing losses and improving animal welfare.

11. FUTURE SCOPE

- Add more disease classes as data becomes available.
- Integrate with mobile app for field use.
- Multi-language support.
- API for integration with farm management software.
- Collaborate for large-scale deployment in agri-tech sector.

12. REFERENCES

- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks
- Poultry disease datasets (cite sources)
- Flask and TensorFlow documentation

13. APPENDIX

Source Code

- GitHub Repository

Dataset Link

- (Insert dataset source or upload location)

GitHub & Project Demo Link

- GitHub Repo
- [Video Demo](video demo/readme.md)

Additional Diagrams

- Data Flow Diagram
- Solution Architecture Diagram
- Confusion Matrix

Video Demo

- (See video demo/readme.md in repo)