AI/ML Project Report

Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management

Team ID: LTVIP2025TMID39070

Team Size: 3

Team Leader: S Susmitha

Team Member: Shaik Saba Naziya

Team Member: Syamagari Surekha

1. INTRODUCTION

1.1 Project Overview

This project aims to create an AI-powered system for detecting and classifying poultry diseases using transfer learning techniques. It utilizes deep learning models trained on generic image datasets and adapts them to poultry-specific visuals, enabling non-invasive diagnosis through image input and delivering actionable guidance to users.

1.2 Purpose

- Support early identification to reduce bird mortality and productivity drops
- Provide accessible diagnostic options in low-resource settings
- Minimize dependence on lengthy lab tests
- Demonstrate transfer learning in agricultural health monitoring
- Contribute to humane and data-driven poultry care

2. IDEATION PHASE

2.1 Problem Statement

Outbreaks in poultry farming can be devastating and spread quickly. Traditional detection often involves subjective assessments or delayed lab results, costing time and leading to greater losses. There's a need for automated, scalable image-based diagnostic tools to aid early intervention.

2.2 Empathy Map Canvas

Sees	Hears	Thinks & Feels	Says & Does	Pains	Gains
Sick or lethargic birds	Vet advice or rumors of nearby illness	Worries about flock health and financial strain	Calls vet, isolates birds, tries remedies	Diagnosis delays, financial loss	Faster, more accurate diagnosis
Unusual droppings, lesions	Farmer discussions and veterinary tips	Frustrated by lack of clarity and speed	Monitors symptoms, researches online	Losses due to wrong decisions	Early intervention, improved flock productivity
Drop in egg/meat output	Supply chain info, disease updates	Hopeful for guidance and help	Takes photos, shares with vet	No direct access to specialists	Better health management and decision support

2.3 Brainstorming

- **Rule-Based Systems**: Easy but rigid
- **Traditional ML**: Simple features, but insufficient for complex visuals
- **CNN from Scratch**: High power, but costly and data-hungry
- **Transfer Learning**: Best balance of accuracy, speed, and practicality

3. REQUIREMENT ANALYSIS

3.1 Customer Journey

1. User opens the app or website
2. Uploads image of affected poultry
3. AI analyzes and returns disease prediction
4. User sees result + confidence score

5. System gives recommended actions
6. User logs result or shares it

3.2 Requirements Tech

Stack:

- Python, Flask
- TensorFlow / Keras
- HTML, CSS
- ResNet50 model
- VS Code as IDE

Core Functionalities:

- Upload and preprocess poultry images
- Predict disease class
- Display result with context and confidence
- Maintain history and enable sharing
- Optional login for security

3.3 Data Flow

---

1. Image upload
2. Storage buffer
3. Preprocessing (resize, normalize)
4. Tensor conversion → passed to ML model
5. Prediction returned
6. Diagnosis details shown on UI
7. Optionally saved in database

3.4 Technology Stack

- ML Frameworks: TensorFlow, PyTorch, Keras
- Image Libraries: OpenCV, Pillow
- Web: Flask/FastAPI, React/Vue
- Cloud: AWS/GCP
- Storage: S3 / Cloud Storage
- Database: PostgreSQL / MongoDB
- Deployment: Docker, Kubernetes
- APIs: RESTful
- Extras: Hugging Face Transformers for future expansion

4. PROJECT DESIGN

4.1 Solution Fit

This system delivers rapid classification of poultry diseases through a smart, image-based tool. It supports early intervention and empowers farmers with actionable data, avoiding delays and costly lab diagnostics.

4.2 Architecture

- **Frontend**: Mobile/web app for uploading poultry images
- **Backend**: Flask API with ResNet50 for inference
- **Pipeline**:
    - Validate image → preprocess → convert to tensor → classify
    - Fetch info → display result → optionally store in DB

Components:

- User Interface
- Image Upload and Preprocessing
- Model Inference Service
- Disease Knowledge Base
- Diagnosis Logger Database
- Sharing & Logging Options

---

5. PLANNING & SCHEDULING

5.1 Agile Timeline

Week 1:

- Set up environments
- Explore dataset (2–3 diseases for MVP) **Week 2**:
- Apply MobileNetV2 or ResNet50
- Run transfer learning **Week 3**:
- Build image upload API
- Connect frontend to backend **Week 4**:
- Cloud deploy, demo test
- Fix bugs, prepare final presentation

---

6. FUNCTIONAL & PERFORMANCE TESTING

6.1 Testing Metrics

- Inference latency
- Upload-to-result speed
- Throughput under load
- CPU/GPU usage
- DB response speed

Testing Tools:

- Locust, JMeter
- TensorFlow profiler
- Prometheus + Grafana
- ELK for logs

8.RESULT

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Poultry Disease Classifier</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap"
rel="stylesheet">
  <style>
    body {
      font-family: 'Inter', sans-serif;
    }
    .result-card {
      transition: all 0.3s ease-in-out;
      transform: translateY(20px);
      opacity: 0;
    }
    .result-card.show {
      transform: translateY(0);
      opacity: 1;
    }
  </style>
</head>
<body class="bg-gray-100 flex items-center justify-center min-h-screen">

  <div class="bg-white rounded-2xl shadow-xl p-8 max-w-lg w-full text-center">
```

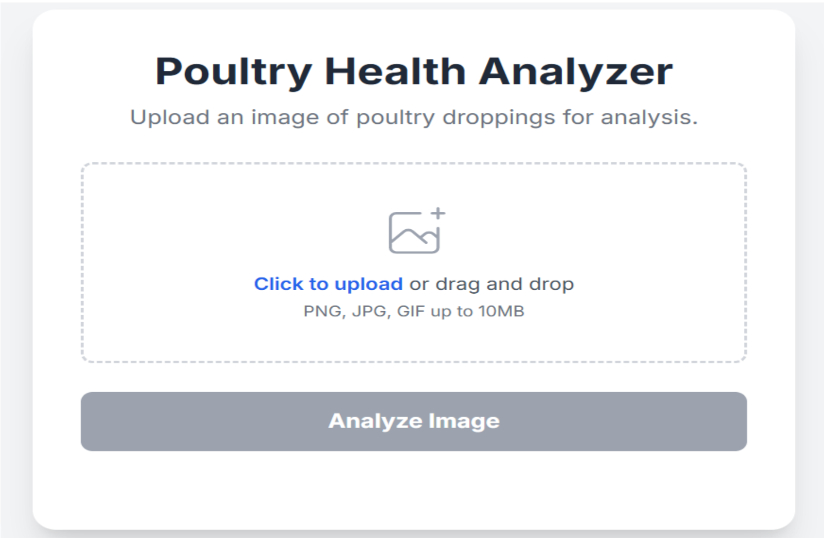
```
    <!-- Header -->
    <h1 class="text-3xl font-bold text-gray-800 mb-2">Poultry Health Analyzer</h1>
    <p class="text-gray-500 mb-6">Upload an image of poultry droppings for analysis.</p>

    <!-- Image Upload Section -->
    <div id="image-uploader" class="border-2 border-dashed border-gray-300 rounded-lg p-8 cursor-pointer hover:border-blue-500 hover:bg-gray-50 transition">
      <input type="file" id="image-upload-input" class="hidden" accept="image/*">
      <div id="upload-prompt">
        <svg class="mx-auto h-12 w-12 text-gray-400" stroke="currentColor" fill="none" viewBox="0 0 48 48" aria-hidden="true">
          <path d="M28 8H12a4 4 0 00-4 4v20m32-12v8m0 0v8a4 4 0 01-4 4H12a4 4 0 01-4-4v-4m32-4l-3.172-3.172a4 4 0 00-5.656 0L28 28M8 32l9.172-9.172a4 4 0 015.656 0L28 28m0 0l4 4m4-24h8m-4-4v8" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" />
        </svg>
        <p class="mt-2 text-sm text-gray-600">
          <span class="font-semibold text-blue-600">Click to upload</span> or drag and drop
        </p>
        <p class="text-xs text-gray-500 mt-1">PNG, JPG, GIF up to 10MB</p>
      </div>
      <img id="image-preview" class="hidden max-h-64 mx-auto rounded-lg" alt="Image preview"/>
    </div>

    <!-- Classify Button -->
    <button id="classify-btn" class="mt-6 w-full bg-blue-600 text-white font-bold py-3 px-4 rounded-lg hover:bg-blue-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-blue-500 transition disabled:bg-gray-400" disabled>
      Analyze Image
    </button>

    <!-- Results Section -->
    <div id="results-container" class="mt-8 text-left">
      <!-- This is where the result card will be injected -->
    </div>

  </div>

  <script>
    const imageUploader = document.getElementById('image-uploader');
    const imageUploadInput = document.getElementById('image-upload-input');
    const uploadPrompt = document.getElementById('upload-prompt');
    const imagePreview = document.getElementById('image-preview');
    const classifyBtn = document.getElementById('classify-btn');
    const resultsContainer = document.getElementById('results-container');
```

```javascript
// Handle clicking the uploader to trigger file input
imageUploader.addEventListener('click', () => imageUploadInput.click());

// Handle file selection
imageUploadInput.addEventListener('change', (event) => {
  const file = event.target.files[0];
  if (file) {
    const reader = new FileReader();
    reader.onload = (e) => {
      imagePreview.src = e.target.result;
      imagePreview.classList.remove('hidden');
      uploadPrompt.classList.add('hidden');
      classifyBtn.disabled = false;
      resultsContainer.innerHTML = ''; // Clear previous results
    };
    reader.readAsDataURL(file);
  }
});

// Handle drag and drop
imageUploader.addEventListener('dragover', (event) => {
  event.preventDefault();
  imageUploader.classList.add('border-blue-500', 'bg-gray-50');
});

imageUploader.addEventListener('dragleave', (event) => {
  event.preventDefault();
  imageUploader.classList.remove('border-blue-500', 'bg-gray-50');
});

imageUploader.addEventListener('drop', (event) => {
  event.preventDefault();
  imageUploader.classList.remove('border-blue-500', 'bg-gray-50');
  const file = event.dataTransfer.files[0];
  if (file) {
    // Manually set the file to the input
    imageUploadInput.files = event.dataTransfer.files;
    // Trigger the change event
    const changeEvent = new Event('change');
    imageUploadInput.dispatchEvent(changeEvent);
  }
});
```

```
    // Handle classification button click
    classifyBtn.addEventListener('click', () => {
      // --- SIMULATION ---
      // In a real application, you would send the image to a backend
      // server running the TensorFlow model. Here, we simulate the response.
      classifyBtn.textContent = 'Analyzing...';
      classifyBtn.disabled = true;

      setTimeout(() => {
        const diseases = [
          { name: 'Coccidiosis', color: 'red-500', description: 'Characterized by bloody or watery
droppings. Immediate isolation and treatment are recommended.' },
          { name: 'Healthy', color: 'green-500', description: 'Droppings appear normal. The bird
seems to be in good health.' },
          { name: 'Newcastle Disease', color: 'yellow-500', description: 'Often indicated by greenish,
watery diarrhea. This is a serious condition requiring vet consultation.' },
          { name: 'Salmonella', color: 'orange-500', description: 'Associated with pasty or brownish
droppings. Biosecurity measures should be checked.' }
        ];

        // Simulate a random prediction
        const prediction = diseases[Math.floor(Math.random() * diseases.length)];
        const confidence = (Math.random() * (0.98 - 0.75) + 0.75).toFixed(2);

        displayResults(prediction, confidence);

        classifyBtn.textContent = 'Analyze Image';
        // Keep button enabled for another analysis
        classifyBtn.disabled = false;

      }, 2000); // Simulate network delay
    });

    function displayResults(prediction, confidence) {
      resultsContainer.innerHTML = `
        <div id="result-card" class="result-card bg-gray-50 rounded-lg p-6">
          <h3 class="text-lg font-semibold text-gray-800 mb-2">Analysis Result</h3>
          <div class="flex items-center justify-between">
            <p class="text-xl font-bold text-${prediction.color}">
              ${prediction.name}
            </p>
            <p class="text-sm font-medium text-gray-600 bg-gray-200 rounded-full px-3 py-1">
              Confidence: ${(confidence * 100).toFixed(1)}%
            </p>
```

```
            </div>
            <div class="w-full bg-gray-200 rounded-full h-2.5 mt-3">
              <div class="bg-${prediction.color} h-2.5 rounded-full" style="width: ${confidence * 100}%"></div>
            </div>
            <p class="text-sm text-gray-600 mt-4">${prediction.description}</p>
          </div>
        `;
        // Trigger the animation
        setTimeout(() => {
          document.getElementById('result-card').classList.add('show');
        }, 100);
      }
  </script>

</body>
</html>
```

OUTPUT :



8.ADVANTAGES & LIMITATIONS

Advantages

- Quick disease detection
- Supports farmers with limited resources
- Minimizes guesswork
- Works with small datasets thanks to transfer learning
- Easy to scale and deploy

Limitations

- Image quality impacts accuracy
- Training benefits from GPU hardware
- Similar symptoms may confuse results
- Model not interpretable out of the box
- Must be paired with vet confirmation

9. CONCLUSION

This system shows how deep learning, especially transfer learning, can revolutionize poultry healthcare. It reduces delay in detection, supports early action, and offers scalability for widespread use in agriculture.

10. FUTURE PLANS

- Track disease progression
- Combine IoT data and visuals
- Add sound/text input support
- Severity grading
- Integrate explainability (Grad-CAM)
- Expand to other livestock