# Step-by-Step Guide: Real-time Cotton Disease Detection System

Here's a complete step-by-step guide to setting up and running your real-time cotton disease detection system:

## Phase 1: Preparation and Setup

### Step 1: Gather Required Materials

- Raspberry Pi 4 with power supply

- Raspberry Pi Camera Module

- MicroSD card (16GB or larger)

- Computer for initial setup

- Internet connection

### Step 2: Install Raspberry Pi OS

1. Download Raspberry Pi Imager from [raspberrypi.com/software](https://www.raspberrypi.com/software/)

2. Insert microSD card into your computer

3. Use Imager to install Raspberry Pi OS (64-bit Lite recommended)

4. Enable SSH and configure WiFi during imaging process

5. Eject microSD card and insert into Raspberry Pi

### Step 3: Initial Raspberry Pi Setup

1. Power on Raspberry Pi

2. SSH into your Pi (find IP using your router admin panel or use `raspberrypi.local`)

```bash

ssh pi@raspberrypi.local

```

Default password: `raspberry`

3. Update system:

```bash

sudo apt update && sudo apt upgrade -y

sudo reboot

```

## Phase 2: Install Dependencies

### Step 4: Install Required Packages

```bash

# Install system dependencies

sudo apt install -y python3-pip python3-opencv libatlas-base-dev libjasper-dev libqtgui4 libqt4-test

# Install Python packages

pip3 install tensorflow opencv-python pillow streamlit fastapi uvicorn requests numpy

```

### Step 5: Enable Camera Interface

```bash

sudo raspi-config

```

- Navigate to "Interface Options"

- Select "Camera" and enable it

- Reboot when prompted

### Step 6: Test Camera

```bash

# Test camera detection

vcgencmd get\_camera

# Take a test photo

raspistill -o test.jpg

```

## Phase 3: Project Setup

### Step 7: Create Project Directory

```bash

mkdir ~/cotton\_disease\_detection

cd ~/cotton\_disease\_detection

```

### Step 8: Transfer Your Model Files

From your computer, transfer these files to the Raspberry Pi:

- `cotton\_disease\_model.h5` (your trained model)

- `class\_names.json` (your class names file)

Using SCP (from your computer):

```bash

scp cotton\_disease\_model.h5 class\_names.json pi@raspberrypi.local:~/cotton\_disease\_detection/

```

### Step 9: Create the Application Files

On your Raspberry Pi, create these files in the project directory:

1. \*\*Create `pi\_camera\_detection.py`\*\* (copy the full code provided earlier)

2. \*\*Create `streamlit\_app.py`\*\* (copy the full code provided earlier)

3. \*\*Create `api\_server.py`\*\* (copy the full code provided earlier)

4. \*\*Create `setup.sh`\*\* (copy the provided setup script)

5. \*\*Create `run\_system.sh`\*\* (copy the provided run script)

Make the scripts executable:

```bash

chmod +x setup.sh run\_system.sh

```

## Phase 4: System Configuration

### Step 10: Configure Network Settings

Ensure your Raspberry Pi has a static IP or you know its current IP:

```bash

hostname -I

```

### Step 11: Test Individual Components

Test the API server:

```bash

python3 api\_server.py

```

Press Ctrl+C to stop it

Test the camera detection:

```bash

python3 pi\_camera\_detection.py

```

Press 'c' to capture, 'q' to quit

## Phase 5: Running the System

### Step 12: Start the Complete System

```bash

./run\_system.sh

```

This will:

1. Start the API server on port 8000

2. Start the Streamlit UI on port 8502

3. Start the camera detection system

### Step 13: Access the Web Interface

On any device connected to the same network:

1. Open a web browser

2. Navigate to: `http://[YOUR\_PI\_IP]:8502`

(Replace [YOUR\_PI\_IP] with your Raspberry Pi's IP address)

### Step 14: Using the System

1. \*\*Live Detection Tab\*\*: View real-time camera feed

2. \*\*Press 'c'\*\* on the Raspberry Pi terminal to capture and analyze an image

3. \*\*View Results\*\*: The web interface will automatically update with:

- Disease detection results

- Sprinkle recommendation (YES/NO)

- Advice in English and Hindi

- Recommended pesticides (if needed)

4. \*\*History Tab\*\*: View previous detections

5. \*\*Disease Information Tab\*\*: Learn about different cotton diseases

## Phase 6: Daily Operation

### Step 15: Regular Use

1. Ensure Raspberry Pi is powered on

2. Run the system: `./run\_system.sh`

3. Access the web interface from your phone, tablet, or computer

4. Point camera at cotton leaves

5. Press 'c' to capture and analyze

6. Follow the recommendations provided

### Step 16: Maintenance

- Regularly update the system:

```bash

sudo apt update && sudo apt upgrade -y

pip3 install --upgrade tensorflow opencv-python pillow streamlit

```

- Monitor system resources:

```bash

htop

```

## Troubleshooting Common Issues

### Camera Not Working

```bash

# Check camera detection

vcgencmd get\_camera

# Check camera permissions

groups pi

```

### TensorFlow Issues

If TensorFlow is too heavy for your Pi, consider:

```bash

# Install lighter version

pip3 uninstall tensorflow

pip3 install tensorflow-lite

```

### Port Already in Use

```bash

# Find processes using ports

sudo lsof -i :8000

sudo lsof -i :8502

# Kill processes if needed

sudo kill -9 [PID]

```

## System Flow Explanation

1. \*\*Camera Capture\*\*: Raspberry Pi camera captures images

2. \*\*Disease Prediction\*\*: TensorFlow model analyzes the image

3. \*\*API Communication\*\*: Results sent to FastAPI server

4. \*\*Web Interface\*\*: Streamlit displays results in real-time

5. \*\*Farmer Advice\*\*: System provides sprinkle recommendations and treatment advice

## Important Notes

1. \*\*Performance\*\*: Raspberry Pi 4 may have slower inference times (~5-10 seconds per image)

2. \*\*Lighting\*\*: Ensure good lighting for accurate detection

3. \*\*Camera Focus\*\*: Adjust camera focus for clear leaf images

4. \*\*Network\*\*: Keep Raspberry Pi connected to power and network

5. \*\*Security\*\*: This setup is for local network use only

This system provides a complete real-time cotton disease detection solution with farmer-friendly advice in both English and Hindi, specifically telling farmers when to sprinkle pesticides and when not to.