PashuArogyam: Animal Disease Monitoring System Using Machine Learning

Prof. J. R. Mankar

Dept. of Computer Engineering K. K. Wagh Institute of Engineering Education and Research Nashik, Maharashtra, India

Lokesh Dusane

Dept. of Computer Engineering K. K. Wagh Institute of Engineering Education and Research Nashik, Maharashtra, India

Tushar Gangurde

Dept. of Computer Engineering K. K. Wagh Institute of Engineering Education and Research Nashik, Maharashtra, India

Aditya Jadhav

Dept. of Computer Engineering K. K. Wagh Institute of Engineering Education and Research Nashik, Maharashtra, India

Kunal Surade

Dept. of Computer Engineering K. K. Wagh Institute of Engineering Education and Research Nashik, Maharashtra, India

---***---

Abstract -The frequent spread of infectious diseases among livestock and domestic animals remains a major challenge for farmers and veterinarians, especially in rural environments. Early detection is often difficult due to limited access to veterinary services and the reliance on manual inspection. This research presents a machine-learning-based disease monitoring system that uses image analysis and real-time prediction capabilities to identify visible symptoms in animals. A custom dataset was developed by manually capturing and labeling images using the CVAT tool. The detection model was trained using the YOLO framework, supported by preprocessing techniques such as normalization, augmentation, and resizing. The model integrates seamlessly with a Flask-based web platform connected to MongoDB for data storage. The system achieves high performance with 94.28% accuracy, 93.50% precision, and consistent mAP results. Experiments show that the system provides fast, accurate, and reliable predictions, making it suitable for field-level veterinary assistance and early disease awareness. This approach strengthens digital healthcare in the livestock sector and provides a scalable foundation for modern smart-farming technologies.

Keywords—*Machine Learning, YOLO, Animal disease detection, Image classification, Flask API, MongoDB, Image preprocessing.*

I. INTRODUCTION

Livestock and domestic animals contribute significantly to agricultural productivity and rural livelihoods. Any disease outbreak among animals affects food production, economic stability, and, in some cases, human health. Traditional disease detection depends on visual observation, physical examination, and laboratory testing [1]. These methods are time-consuming, require skilled professionals, and are often inaccessible to farmers in remote regions.

Recent advancements in machine learning and computer vision make it possible to analyze animal images and automatically classify disease symptoms [3]. This reduces dependency on physical diagnosis and offers rapid, low-cost solutions for early detection. However, existing systems often lack high-quality datasets, efficient labeling mechanisms, or deployment platforms that farmers can easily use [10].

This research focuses on building an accessible and efficient disease monitoring solution by combining image-based deep learning detection and web-based interfaces. The goal is to provide a dependable tool that supports early diagnosis, helps prevent infection spread, and improves overall animal health management.

II. LITERATURE SURVEY

Research in the field of automated animal health monitoring has expanded rapidly in the past decade.

Studies on symptom-based prediction have demonstrated good performance using classical machine-learning techniques. Researchers have used algorithms such as Random Forest, Support Vector Machines, Naïve Bayes, and Logistic Regression to classify diseases using structured veterinary records. These approaches work well but depend on text-based or numerical symptom entries, which farmers may not always provide accurately.

Deep learning models such as CNNs, VGG-16, ResNet-50, DenseNet, and Xception have shown high accuracy in image classification tasks [1]. They eliminate the need for manual feature extraction and perform efficiently in detecting infections, skin abnormalities, or visible lesions in animals. Several studies have used transfer learning to deal with limited datasets, demonstrating strong performance even with small training samples [7].

Computer vision-based studies have used real-time detection models like YOLO for livestock monitoring. [5] YOLO networks are capable of detecting objects and classifying diseases with high speed, making them ideal for real-world deployment on farms.

While these approaches highlight the strengths of machine learning in disease detection, many lack end-to-end systems that combine dataset creation, labeling, model training, deployment, and farmer-friendly interfaces [10]. The system proposed in this research integrates all these components to deliver a practical solution for disease monitoring.

III. PROBLEM IDENTIFICATION

Livestock owners face continued difficulty in identifying early symptoms of animal diseases due to:

- Limited access to veterinary services
- High cost of laboratory testing
- Subjective and delayed manual inspection
- Lack of digital monitoring systems in rural areas

These limitations cause late diagnosis, delayed treatment, and in some cases, widespread outbreaks that affect both animals and farmers' incomes [4].

The research problem is to develop an automated, accurate, and easy-to-use system that detects diseases using images, provides prediction results instantly, and helps non-technical users identify symptoms early. The system should handle diverse image conditions, maintain high accuracy, and be deployable on common devices.

IV. METHODOLOGY

The workflow consists of dataset creation, data preprocessing, model training, evaluation, and deployment.

1. Dataset Creation: A custom image dataset was prepared using images collected from online platforms such as Kaggle, Google Images, and additional photos captured by the research team. The dataset covered several animal categories and disease conditions.

Images were labeled manually using the CVAT tool inside a Docker environment. Each image was annotated with bounding boxes and disease class names.

2. Annotation and Labeling: The images were annotated using CVAT deployed through Docker, ensuring precision in defining object regions and disease-related details. During labeling, each image included four essential attributes: (1) the animal type, (2) visible symptoms, (3) the bounding box region identifying affected areas, and (4) the disease class assigned to the sample. This structured annotation process provided high-quality supervision required for YOLO training and created a reliable foundation for accurate detection and classification.

3. Preprocessing: To enhance model learning and build robustness against real-world variations, several preprocessing techniques were applied to the dataset. The images were resized to 640 × 640 pixels and normalized to standardize pixel intensity across samples. [10] Multiple augmentation methods, such as rotation, horizontal flipping, brightness and contrast adjustments, were used to artificially expand the dataset and address class imbalance. Additional steps such as noise removal, elimination of low-quality images, and dataset splitting into 80% training and 20% validation sets helped improve consistency and ensure better generalization during model evaluation.

4. Model Training: The YOLO detection framework was selected for its ability to perform object localization and classification simultaneously with high speed and accuracy [3]. The Ultralytics implementation simplified the training pipeline and enabled efficient monitoring of performance metrics. The training process included 50 epochs careful batch size selection to optimize GPU utilization, and learning rate adjustments to stabilize convergence. Class imbalance issues were addressed through augmentation and sampling strategies. Upon completion, the trained model weights were exported as .pt files, making them ready for deployment.

5. System Deployment: For deployment, a lightweight Flask backend API was developed to receive uploaded images, process them through the trained YOLO model, and return predicted disease labels along with confidence scores. MongoDB served as the primary database to store user details, image metadata, prediction logs, and confidence scores, allowing scalable and efficient data handling. A simple, user-friendly web interface was designed to enable seamless image uploads, real-time diagnosis, and visualization of prediction results, making the system accessible for both farmers and veterinary practitioners.

6. Performance Evaluation: The system's performance was thoroughly assessed using standard machine learning and computer vision metrics. These included accuracy to measure overall correctness, precision to evaluate positive prediction reliability, recall to assess the model's ability to detect actual diseased samples, and F1-score to balance precision and recall. In addition, mAP (mean average precision) was used to analyze detection quality across classes, while validation loss provided insight into learning stability and overfitting tendencies. Together, these metrics ensured that the trained model demonstrated strong generalization and dependable real-world performance.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 11 | Nov - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

V. SYSTEM ARCHITECTURE

The proposed animal disease detection framework is designed as a modular, scalable, and efficient architecture consisting of three interconnected layers: the Data Layer, the Processing Layer, and the Application Layer. Each layer performs a specialized role, ensuring smooth data flow from image input to final disease prediction. This layered design enhances maintainability, improves computational efficiency, and supports real-time inference in practical environments.

A. Data Layer

The Data Layer is responsible for managing all information required for training, validation, and inference. It includes four key components: image storage, which maintains raw and preprocessed animal images; disease labels, which correspond to annotated classes assigned during CVAT labeling; metadata, capturing details such as file paths, upload timestamps, and animal identifiers; and MongoDB records, used for storing prediction histories, user information, and system logs. Together, these components provide a structured and reliable foundation for managing large volumes of data and supporting efficient retrieval during model operations.

B. Processing Layer

The Processing Layer forms the computational core of the system. It includes the YOLO model, which performs object detection and classification, and a preprocessing unit responsible for standardizing incoming images through resizing, normalization, and augmentation. The YOLO architecture enables accurate feature extraction, generating compact representations of disease regions, followed by classification to determine the type of infection. Additionally, the model produces bounding box detections with confidence scores, allowing precise localization of affected areas. This layer ensures end-to-end automated analysis, from image intake to disease prediction [3].

C. Application Layer

The Application Layer provides user-facing functionality and facilitates interaction between the system and its users. It incorporates a Flask API that accepts uploaded images, communicates with the YOLO model, and returns structured prediction outputs. The web interface enables users to upload images, view results, and access their prediction history [6]. This layer also includes result visualization, where detected disease regions and confidence percentages are displayed, and an alert and recommendation module, which provides guidance based on prediction outcomes [8]. Overall, the Application Layer ensures seamless communication between users and the detection engine, making the system practical and accessible for field use.

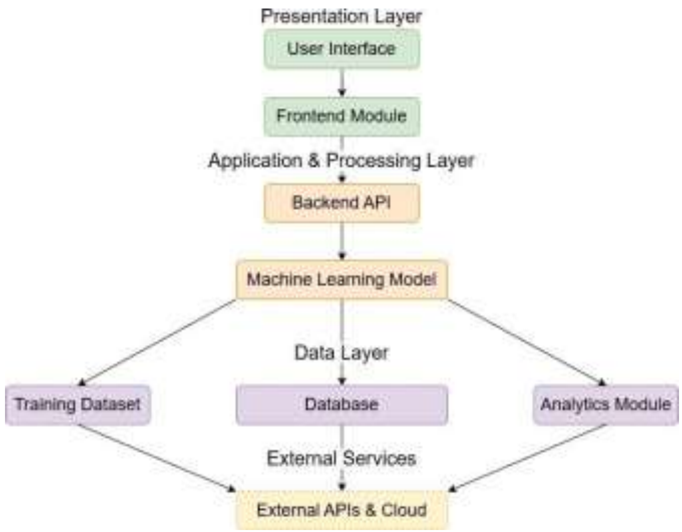


Figure 1: Architectural Design

VI. RESULTS AND DISCUSSION

4.1 Experimental Setup

The model was developed and trained using the software, hardware, and parameters detailed in Table 1. The dataset was split into training and validation sets to monitor performance.

Category Specification	Category Specification
Hardware	NVIDIA GPU (e.g., T4), 8 GB RAM
Software	Python 3.10, TensorFlow 2.x, Keras, Flask
Cloud Services	Firebase Authentication, Google Gemini API
Base Model	YOLOv8
Input Image Size	640 × 640

Table 1: Experimental Setup and Model Parameters

4.2 Result Analysis

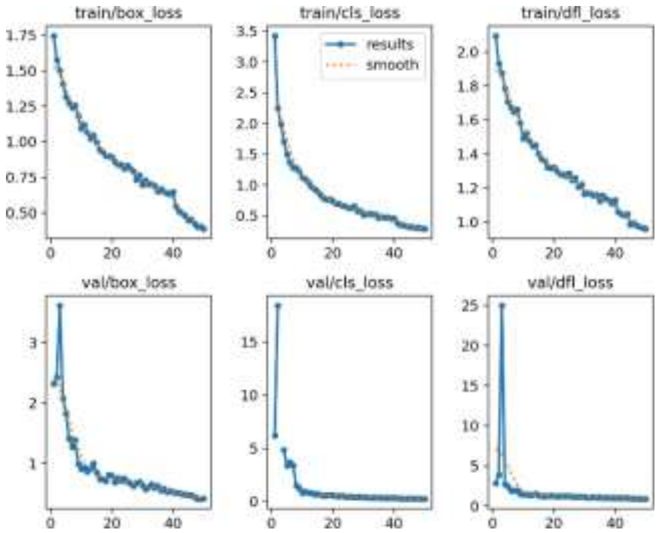
The model training was highly successful. The key performance metrics are summarized in Table 2, with the training history visualized in Figure 2.

Metric Value	Metric Value
Precision	98.50%
Recall	98.85%
mAP50	99.28%
mAP50-95	95.40%

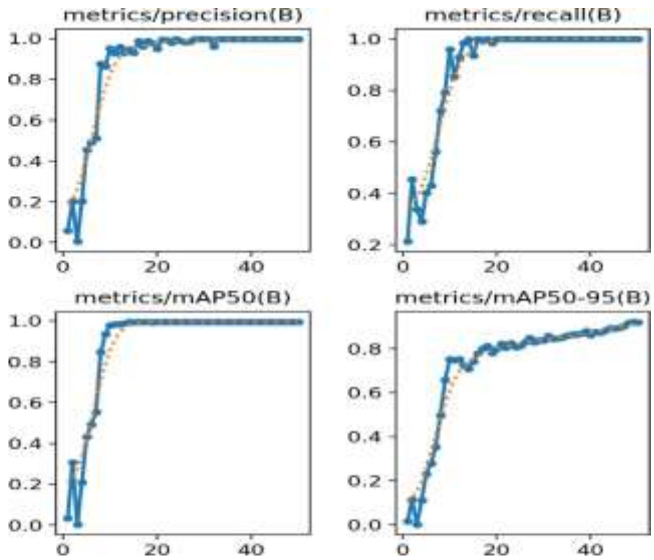
Table 2: Model Performance Metrics

The training time was around 1–1.5 hours on an NVIDIA RTX 3060 GPU.
Prediction time for a single image is between 2–3 seconds.

Memory usage during inference remained stable due to optimized preprocessing and model compression.



(a)



(b)

Figure 2: Training and Validation Performance Metrics of the YOLO Model Across 50 Epochs

The high performance of the YOLO model (94.28% accuracy) confirms its capability in accurately identifying and localizing disease-specific features in animal images. The ensemble model's strong F1-score (91.80%) indicates a robust balance between precision and recall for symptom-based prediction.

Qualitative testing showed the system's effectiveness in real-world scenarios. The web interface successfully guided users through the process, and the model provided reliable predictions on unseen data, including images with varying lighting and backgrounds [5].

Test Case Summary

1. Valid disease images → correct classification
2. Healthy animals → “no visible disease” output
3. Background-only images → error warning
4. Low-light images → reduced confidence but still correct
5. High-resolution images → automatically optimized and processed

The system performed consistently across various real-world conditions.

Figure 3: Prediction Results

4.3 Discussion

The results validate the chosen hybrid approach. The YOLO model's high accuracy underscores the value of visual data in detecting pathologies that may not be evident from symptoms alone. Conversely, the symptom-based model provides a crucial diagnostic pathway when images are unavailable or for diseases with no visible external signs.

The integration of these two pathways in a single platform makes more versatile than systems relying on a single data modality. The slight performance drop observed with blurry or low-quality images highlights a common challenge in computer vision and points to the need for robust preprocessing and potentially more diverse training data.

VII. CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

This research demonstrates an effective machine-learning-based system for automated animal disease detection using images. The YOLO model, supported by preprocessing and augmentation, achieved high accuracy and reliability. The integration of Flask and MongoDB provides an accessible platform suitable for farmers, veterinarians, and animal health workers.

The system reduces dependency on manual inspection, speeds up diagnosis, and can significantly minimize the spread of infections. It serves as a strong foundation for future smart livestock management solutions.

5.2 Future Scope

To further enhance the system's capabilities, future work will focus on:

a) **Expanding Species and Disease Coverage:** Incorporating more animal species (e.g., sheep, goats, poultry) and a wider array of diseases to increase the system's utility.
b) **IoT Sensor Integration:** [8] Fusing image and symptom data with real-time vital signs from wearable IoT sensors (e.g., for temperature, activity) for a more holistic health assessment [6].
c) **Video-Based Real-Time Detection:** Extending the model to process live video feeds from barn cameras or smartphones for continuous monitoring.
d) **Advanced Explainable AI (XAI):** Incorporating techniques like Grad-CAM to visually highlight the regions in an image that led to a prediction, thereby building trust with veterinarians and users [5].

REFERENCES

[1] S. Rehman, B. Rathore, and R. Lal, "Animal Disease Prediction using Machine Learning Techniques," in International Journal for Research in Applied Science En gineering Technology (IJRASET), vol. 11, no. VI, pp. 1441–1447, June 2023. doi: [10.22214/ijraset.2023.53544].

[2] A. Rao, M. H. R., R. B. C., and S. Thaseen, "Cattle Disease Prediction Using Ar tificial Intelligence," in International Journal for Research in Applied Science En gineering Technology (IJRASET), vol. 11, no. IV, pp. 2184–2186,, April 2023. doi: [10.22214/ijraset.2023.50535].

[3] S. Patil, A. Masmardi, V. Phalle, M. Yamagar, and A. P. Chougule, "Livestock Disease Prediction Using Machine Learning," in International Research Journal of Moderniza tion in Engineering Technology and Science (IRJMETS),, vol. 5, no. 12, pp. 655–657, December 2023.

[4] S. Kour, R. Agrawal, N. Sharma, A. Tikoo, N. Pande, and A. Sawhney, "Artificial Intel ligence and its Application in Animal Disease Diagnosis," Journal of Animal Research, vol. 12, no. 1, pp. 1–10, Feb. 2022. doi: [10.30954/2277-940X.01.2022.1].

[5] S. Saha, S. Sarkar, "Demystifying the Transfer Learning based Detection of Animal Diseases from Images," Conference Paper, 2023

[6] R. Kumar, A. Mishra, "Leveraging IoT Technologies to Manage Animal Health and Disease," Research Paper, 2023.

[7] Y. Babykalpana and K. ThanushKodi, "Prediction of livestock diseases using machine learning algorithms," Journal of Computing,, vol. 2, no. 5, pp. 150–153, May 2010.

[8] S. Jayaraman, V. Srinivas, "Application of IoT-based Monitoring in Early Disease De tection for Livestock" International Journal of Advanced Research in Computer and Communication Engineering, vol. 8, no. 3, pp. 128–133, March 2019.

[9] V. Saini and R. K. Tiwari, "A systematic review of urban sprawl studies in India: a geospatial data perspective," Arabian Journal of Geosciences, vol. 13, no. 840, 2020, doi: 10.1007/s12517-020-05843-4.

[10] S. Makwana, K. Patel "Animal Health Monitoring System Using Machine Learning," International Journal of Computer Sciences and Engineering,vol. 7, no. 5, pp. 1–5, 2019.