# IMPLEMENTATION OF DISEASE PATTERN RECOGNITION & CARETAKING SYSTEM FOR ANIMALS

**A MINOR PROJECT SYNOPSIS**

***Submitted in partial fulfillment of the requirements for the award of the degree***

***Of***

**BACHELOR OF TECHNOLOGY**

***In***

**ELECTRONICS & COMMUNICATION ENGINEERING**

***By***

**NISSAN SUNWAR**
**06515602821**
**F-4**
**+91 8287590032**

***Guided by***

**Mr. Varun Jain , Assistant professor. , ECE Department**

**Department of Electronics & Communication Engineering**

**DR. AKHILESH DAS GUPTA INSTITUTE OF PROFESSIONAL STUDIES**

**(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY)**

**NEW DELHI – 110053**

**August- 2024**

# CONTENTS

# INTRODUCTION

In the modern world, where pet ownership and livestock management are integral to many households and industries, the early detection and effective management of animal diseases are crucial. However, diagnosing animal diseases often requires specialized knowledge and access to veterinary services, which may not always be readily available. To address this challenge, we propose a machine learning-based system designed to recognize animal diseases and provide caretaking recommendations. This system leverages advances in artificial intelligence, image recognition, and natural language processing to assist pet owners, farmers, and veterinarians in diagnosing diseases and offering appropriate care.

# OBJECTIVE

Developing a machine learning model to identify diseases in animals based on symptoms, images, or other diagnostic data and provide caretaking recommendations. Deciding whether the model will focus on a specific group of animals (e.g., pets like dogs and cats, livestock like cows and pigs, or wildlife) or be more generalized.

Specify the diseases to be included, such as common infections, genetic disorders, nutritional deficiencies, etc.
For caretaking recommendations define the type of advice the model will provide, such as veterinary consultation, dietary changes, isolation procedures, or medication suggestions.

In Fig. 1, the overall flow of the System, The model will continuously monitor a cow grazing in the field through a camera that track vital signs like eating patterns, body temperature, and activity levels. This will wirelessly transmit the collected health data in real-time to a central device or cloud platform. The AI model on this platform will analyze the data to detect any signs of illness or abnormal behavior, sending alerts to the farmer or veterinarian if intervention is needed, ensuring timely care and maintaining the cow's health.


Fig.1: Schematic Overview & Outcomes of the System

# BLOCK DIAGRAM

In Fig. 2 the stages involved in training and executing the deep learning model are demonstrated & the workflow starts with input data collection (splitting training set & test set), followed by data pre-processing. The processed data is then used to train the Deep Learning Classification Model. Then, the model undergoes Hyperparameter Tuning, followed by Testing and Validation as a final model-building stage in order to increase the accuracy further. Once the model is trained, it is deployed for real-time prediction.

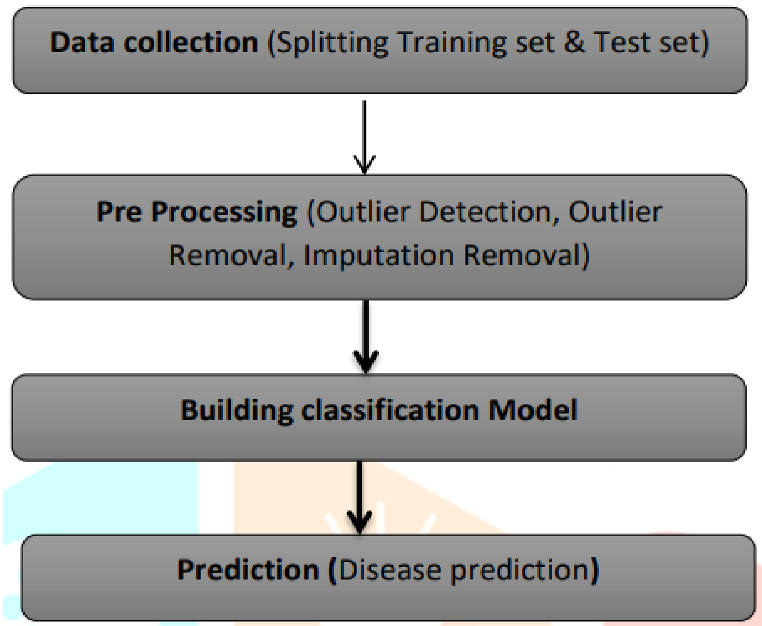


Fig. 2: Block Diagram of Deep Learning Model Workflow

# METHODOLOGY & TOOLS USED

## Data Collection

1. Imagery Data: High-quality images of affected areas (skin lesions, eyes, etc.).
2. Textual Data: Descriptions of symptoms, medical history, and environmental factors.
3. Sensor Data: If applicable, include data from wearables (e.g., temperature, heart rate).
4. Diagnostic Results: Lab tests, X-rays, bloodwork.

Sources for Data Collection :

1. Veterinary Clinics: Collaborate with veterinary clinics to obtain labeled datasets.
2. Research Databases: Use existing datasets from veterinary research.
3. Crowdsourcing: Develop an app for pet owners to upload data (with veterinary confirmation).
4. Data Labelling: Ensure data is accurately labeled with the correct disease diagnosis and related caretaking advice.

## Data Preprocessing

1. Data Cleaning: Remove duplicates, handle missing data, and correct any inconsistencies.
2. Data Augmentation: For images, use techniques like rotation, flipping, and zooming to increase the dataset size.
3. Normalization: Normalize data (e.g., scaling images, standardizing symptom descriptions).
4. Feature Engineering: Extract relevant features (e.g., color, shape, texture from images; keyword extraction from text).

# Model Development

## Model Selection:

1. Image Classification: Use convolutional neural networks (CNNs) for disease recognition from images.
2. Text Analysis: Employ natural language processing (NLP) models like transformers for analysing textual descriptions of symptoms.
3. Multimodal Models: Combine image and text data using multimodal learning approaches.

## Training and Validation:

1. Split Data: Divide the data into training, validation, and test sets.
2. Model Training: Train the model using appropriate loss functions and optimizers.
3. Validation: Use cross-validation to tune hyperparameters and avoid overfitting.
4. Evaluation Metrics: Accuracy, precision, recall, F1-score, and confusion matrix for classification tasks.

## Caretaking Recommendation System:

1. Rule-Based System: Start with a simple rule-based system that suggests caretaking steps based on the recognized disease.
2. Machine Learning Integration: Use models like decision trees or recommendation algorithms to suggest tailored caretaking actions based on individual cases.
3. Expert Feedback: Involve veterinarians to refine and validate the recommendations.

<u>Model Deployment:</u>

1. Deployment Platform: Choose a platform (e.g., web app, mobile app) where users can upload data and receive diagnoses and advice.
2. API Development: Develop APIs to allow easy integration with other veterinary systems or apps.
3. User Interface: Design a user-friendly interface where users can input data, view results, and get advice.

<u>Testing and Validation:</u>

1. Beta Testing: Release a beta version of the app to a small group of users for feedback.
2. Performance Monitoring: Track the model's performance in the real world and adjust based on false positives/negatives and user feedback.
3. Continuous Improvement: Regularly update the model with new data and retrain it to improve accuracy and recommendation quality.

# Hardware for Automated Feeding System

For automated feeding and caretaking, a combination of sensors, IoT devices, and robotics can be integrated with the AI model.

## Smart Feeders

<u>Purpose</u>: Automate the dispensing of food and water based on the AI model’s recommendations, ensuring proper nutrition and hydration.

<u>Hardware Components:</u>

1. Microcontrollers: Arduino or Raspberry Pi to control the feeding mechanism.
2. Sensors: Weight sensors to measure the amount of food dispensed, and proximity sensors to detect the presence of the animal.
3. Actuators: Motors to control the release of food and water

.

## Environmental Sensors

Purpose: Monitor the living conditions of animals (e.g., temperature, humidity, light) and adjust settings to maintain a healthy environment.

Hardware Components:

1. Temperature and Humidity Sensors: DHT11 or DHT22 to monitor environmental conditions.
2. Light Sensors: LDR (Light Dependent Resistor) for adjusting light levels.
3. Air Quality Sensors: MQ135 for detecting harmful gases and ensuring air quality.

## Health Monitoring Devices

Purpose: Continuously monitor the animal’s health metrics such as heart rate, temperature, and activity levels.

Hardware Components:

1. Wearable Sensors: FitBark or Whistle for monitoring vital signs.
2. RFID Tags: For identifying and tracking individual animals in a larger group.
3. Infrared Cameras: For non-invasive temperature monitoring.

## IoT Gateways

Purpose: Connect all sensors and devices to the cloud, enabling real-time monitoring and control via the AI model.

Hardware Components:

1. IoT Gateway Devices: Raspberry Pi or dedicated IoT hubs like AWS IoT Core.
2. Communication Modules: Wi-Fi, Bluetooth, or Zigbee modules for wireless communication between devices.

# Tools Used

## Data Collection

1. Web Scraping: Python libraries like BeautifulSoup and Scrapy for extracting data from online sources.
2. Mobile App Development: Flutter or React Native for building apps that allow users to submit data.
3. Database Management: PostgreSQL (SQL) or MongoDB (NoSQL) for storing and managing large datasets.

## Data Preprocessing

1. Data Cleaning and Preprocessing: Pandas, NumPy, and scikit-learn for handling data preprocessing tasks.
2. Image Augmentation: TensorFlow, Keras, or OpenCV for applying augmentation techniques to image data.
3. NLP Preprocessing: spaCy or NLTK for processing textual data and extracting relevant features.

## Model Development

1. Deep Learning Frameworks: TensorFlow and PyTorch for developing and training deep learning models.
2. Image Classification: Pre-trained CNN architectures like ResNet, VGG, or Inception for recognizing diseases from images.
3. NLP Models: BERT, GPT, Lamma, or Transformers from Hugging Face for analyzing symptom descriptions and other textual inputs.
4. Model Evaluation: scikit-learn for calculating performance metrics such as accuracy, precision, recall, and F1-score.

## Caretaking Recommendation System

1. Rule-Based Systems: Simple decision logic in Python or using tools like Drools for implementing rules.
2. Recommendation Algorithms: scikit-learn for decision trees and XGBoost for creating personalized recommendations.
3. Expert Systems: Prolog or similar platforms for more advanced decision-making processes.

### Model Deployment

1. Web Development: Django or Flask for creating web services.
2. Mobile App Integration: React Native or Flutter for deploying the mobile interface.
3. Cloud Deployment: AWS, Google Cloud, or Microsoft Azure for scalable infrastructure.
4. API Development: FastAPI or GraphQL for building APIs to interact with the model.

### Testing and Validation

1. Testing Frameworks: pytest for unit testing, Selenium for automated testing of web interfaces.
2. Performance Monitoring: Prometheus and Grafana for tracking system performance in real-time.
3. A/B Testing: Tools like Optimizely or Google Optimize for comparing different model versions.

# Advantages and Applications

### Early Detection and Diagnosis:

Advantage: The model can recognize early signs of diseases in animals, allowing for timely intervention. This can prevent the progression of illnesses and reduce mortality rates.
Application: Pet owners can use the system to identify potential health issues in their pets before they become serious, while farmers can monitor livestock health to avoid widespread disease outbreaks.

### Accessibility to Veterinary Knowledge:

Advantage: The model provides access to veterinary expertise, even in remote or underserved areas where veterinary services may not be readily available.
Application: Farmers in rural areas can use the system to diagnose livestock diseases without needing immediate access to a veterinarian. Pet owners can also get initial guidance before seeking professional care.

## Time and Cost Efficiency:

Advantage: The system reduces the time and cost associated with veterinary consultations by offering preliminary diagnoses and caretaking advice, potentially minimizing unnecessary trips to the vet.
Application: Pet owners can use the model to determine if a veterinary visit is necessary, saving time and money on consultations. Livestock owners can quickly assess the health of multiple animals, streamlining farm operations.

## Improved Animal Welfare:

Advantage: By enabling prompt diagnosis and treatment, the model enhances overall animal welfare, reducing suffering and improving the quality of life for animals.
Application: Animal shelters and rescue organizations can use the system to monitor the health of large numbers of animals and ensure they receive timely care.

## Personalized Caretaking Recommendations:

Advantage: The model provides tailored caretaking advice based on specific symptoms and conditions, ensuring that animals receive appropriate care suited to their needs.
Application: Pet owners can receive customized advice on diet, environment, and care practices based on their pet's specific condition, leading to better health outcomes.

## Scalability and Adaptability:

Advantage: The model can be scaled to include more species, diseases, and conditions, making it a versatile tool that can be adapted to different contexts and needs.
Application: The system can be expanded to cover a wide range of animal species, from pets to livestock to wildlife, making it applicable in various industries such as agriculture, conservation, and pet care.

## Continuous Learning and Improvement:

Advantage: The model can learn from new data and user feedback, continuously improving its accuracy and reliability over time.

Application: As more users interact with the system, it can be refined to provide even more accurate diagnoses and recommendations, ensuring it stays up-to-date with the latest veterinary knowledge.

**Integration with Other Systems:**

Advantage: The model can be integrated with other veterinary tools, IoT devices, and telemedicine platforms, creating a comprehensive ecosystem for animal health management.

Application: Farmers can integrate the model with wearable devices on livestock to monitor health metrics in real-time, while veterinarians can use it as part of a telemedicine service for remote consultations.

# MAJOR PROBLEMS

## 1. Data Quality and Availability

Issue: The accuracy of the model heavily depends on the quality and diversity of the data used for training. Incomplete, biased, or low-quality data can lead to incorrect diagnoses and recommendations.

Challenge: Acquiring large, labeled datasets for different species, breeds, and diseases can be difficult. For some rare diseases or species, data might be scarce or non-existent.

Impact: Poor data quality can result in reduced model accuracy, misdiagnosis, and potentially harmful recommendations.

## 2. Generalization Across Species and Breeds

Issue: Animals of different species and breeds can exhibit varying symptoms for the same disease, making it challenging for a single model to generalize across a wide range of animals.

Challenge: Developing a model that accurately diagnoses diseases across diverse species and breeds requires extensive data and sophisticated algorithms capable of understanding these variations.

Impact: A model that fails to generalize well may perform poorly when applied to species or breeds that were underrepresented in the training data.

## 3. Model Interpretability and Transparency

Issue: AI models, particularly deep learning models, are often seen as "black boxes" because their decision-making processes are not easily interpretable.

Challenge: Veterinarians and users may require clear explanations for why the model made a specific diagnosis or recommendation. This is especially important in critical cases where understanding the reasoning is necessary.

Impact: Lack of transparency can lead to distrust in the system, limiting its adoption by professionals and end-users.

## 4. Overfitting and Underfitting

Issue: Overfitting occurs when the model performs well on training data but poorly on unseen data, while underfitting happens when the model is too simple to capture the underlying patterns.

Challenge: Balancing the model complexity to ensure it generalizes well to new data without overfitting is a common issue in machine learning.

Impact: Overfitting can result in poor performance in real-world scenarios, while underfitting can lead to inaccurate predictions and recommendations.

## 5. Maintenance and Continuous Learning

Issue: The model needs to be regularly updated with new data and retrained to incorporate the latest veterinary knowledge and adapt to emerging diseases.

Challenge: Maintaining and updating the model requires ongoing resources, including access to new data, computational power, and expert input.

Impact: Without regular updates, the model may become outdated, reducing its accuracy and relevance over time.

# FUTURE ASPECTS

1. **Expand to More Species/Diseases:** Gradually increase the scope to include more species or a wider range of diseases.
2. **Integrate with IoT Devices:** Incorporate data from wearables or other IoT devices for real-time monitoring.
3. **Integration with Smart Farming**: The model could be integrated into broader smart farming ecosystems, coordinating with other IoT devices like automated feeders, waterers, and environmental sensors to create fully autonomous livestock management systems.
4. **Real-Time Decision Making**: As IoT devices become more powerful and widespread, the model could enable real-time decision-making and automated interventions, such as administering medication or adjusting environmental conditions immediately when health anomalies are detected.
5. **Scalable Cloud-Based Platforms:** The model could evolve into a scalable cloud-based platform that supports large-scale operations across multiple farms or wildlife reserves, offering centralized monitoring and data analysis for animal health on a global scale.

# REFERENCES & RELATED RESEARCH PAPERS

1. Badillo S., Banfai B., Birzele F., Davydov I.I., Hutchinson L., Kam-Thong T., Siebourg-Polster J., Steiert B. & Zhang J.D. (2020). – An introduction to machine learning. Clin. Pharmacol. Ther., 107 (4), 871–885. https://doi.org/10.1002/cpt.1796
2. Roth J.A., Battegay M., Juchler F., Vogt J.E. & Widmer A.F. (2018). – Introduction to machine learning in digital healthcare epidemiology. Infect. Control Hosp. Epidemiol., 39 (12), 1457–1462. https://doi.org/10.1017/ice.2018.265

3. Sarker I.H. (2021). – Machine learning: algorithms, real-world applications and research directions. SN Comput. Sci., 2 (3), 160. https://doi.org/10.1007/s42979-021-00592-x
4. Lo Vercio L., Amador K. […] & Forkert N.D. (2020). – Supervised machine learning tools: a tutorial for clinicians. J. Neural Eng., 17 (6), 062001. https://doi.org/10.1088/1741-2552/abbff2
5. Zhou Y.-H. & Gallins P. (2019). – A review and tutorial of machine learning methods for microbiome host trait prediction. Front. Genet., 10, 579. https://doi.org/10.3389/fgene.2019.00579
6. Scott I.A. (2021). – Demystifying machine learning: a primer for physicians. Intern. Med. J., 51 (9), 1388–1400. https://doi.org/10.1111/imj.1520
7. Ferro C., López M., Fuya P., Lugo L., Cordovez J.M. & González C. (2015). – Spatial distribution of sand fly vectors and ecoepidemiology of cutaneous leishmaniasis transmission in Colombia. PLoS One, 10 (10), e0139391. https://doi.org/10.1371/journal.pone.0139391
8. Altman N. & Krzywinski M. (2017). – Clustering. Nat. Methods, 14 (6), 545–546. https://doi.org/10.1038/nmeth.4299
9. Warns-Petit E., Morignat E., Artois M. & Calavas D. (2010). – Unsupervised clustering of wildlife necropsy data for syndromic surveillance. BMC Vet. Res., 6, 56. https://doi.org/10.1186/1746-6148- 6-56.
10. Detecting Disease In Livestock and Recommending Medicines using Machine Learning | IEEE Conference Publication
11. Application of Machine Learning in Animal Disease Analysis and Prediction
12. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8754556
13. https://github.com/thyagarajank/Cattle-disease-prediction-using-Machine-Learning/tree/main