Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 2 (2024)

Integration of Mathematical Operators Based on Decision Boundary Complexity and Combinatorial Optimization for Improved Deep Learning Classifiers

Shruti Thapar[1], Krati Sharma[2], Dharmveer Yadav[3], Budesh Kanwer[4], Ashish Raj[5]

[1]Department of Electronics & Communication, Poornima Institute of Engineering & Technology, Jaipur, Rajasthan, India

[2]Department of English and Soft Skills, Poornima Institute of Engineering & Technology, Jaipur, Rajasthan, India

[3]Department of Computer Science, St. Xavier's College, Jaipur, Rajasthan, India

[4]Department of AI&DS, Poornima Institute of Engineering & Technology, Jaipur, Rajasthan, India

[5]Department of Electrical and Electronics Engineering, Poornima University, Jaipur, Rajasthan, India

Email- shruti.thapar@poornima.org [1], kratibhomia@gmail.com [2], dharmveerya21@gmail.com [3], budesh82@gmail.com [4], ashishraj1987@gmail.com [5]

Article History:

Received: *16-02-2024*

Revised: *22-04-2024*

Accepted: *05-05-2024*

Abstract:

The proliferation of complex diseases in livestock, such as lumpy skin disease, demands advanced diagnostic tools that can accurately classify and predict outbreaks. This study explores the integration of complex mathematical operators within deep learning classifiers to enhance their accuracy and efficiency in diagnosing lumpy skin disease. By focusing on the decision boundary complexity, which delineates different disease states in high-dimensional spaces, and employing combinatorial optimization techniques, we develop a novel framework that significantly improves classification performance. The methodology hinges on optimizing the configuration and combination of mathematical operators, such as gradient operators and higher-order derivatives, to refine feature extraction processes. This approach allows for a more nuanced understanding of the disease features that are critical for accurate classification. Using a dataset comprised of clinical and image data from infected cattle, our enhanced classifiers demonstrate a marked improvement in predictive accuracy compared to traditional deep learning models. The case study not only underscores the potential of integrating advanced mathematical concepts into deep learning but also sets a precedent for tackling similar challenges in veterinary medicine and beyond.

Keywords: Deep Learning Classifiers, Lumpy Skin Disease, Decision Boundary Complexity, Combinatorial Optimization, Mathematical Operators, Feature Extraction, Livestock Disease Classification, Predictive Accuracy, Veterinary Medicine, High-Dimensional Data Analysis.

1. MATHEMATICAL FOUNDATIONS AND OPTIMIZATION TECHNIQUES:

Deep learning, a subset of machine learning, has revolutionized numerous fields from image and speech recognition to autonomous driving and medical diagnosis. At its core, deep learning constructs algorithms, known as neural networks, that can learn and make intelligent decisions on their own. This ability stems from the network's capacity to perform complex mathematical operations and optimize these operations for better performance and accuracy. Understanding these underlying mathematical concepts and optimization techniques is crucial for advancing deep learning applications.

Mathematical Operators in Neural Networks

The foundation of any neural network is its ability to perform mathematical calculations on input data to extract features and make predictions. These calculations are primarily linear algebraic operations, which involve vectors, matrices, and tensors. At the basic level, a neural network transforms its input through a series of layers, each defined by a set of weights and biases, and applies nonlinear transformations to enable complex representations.

1. **Linear Transformations and Decision Boundaries**: Every neural network layer essentially performs a linear transformation followed by a nonlinear activation. The simplest form of a linear transformation in a binary classifier can be represented by the decision boundary equation $f(x)=wTx+b=0$ .This equation defines a hyperplane that divides the input space into two decision areas, each corresponding to one of the classes.
2. **Optimization Techniques**: Optimization is at the heart of training deep learning models. It involves adjusting the weights of the network to minimize a loss function, which measures the difference between the predicted output and the actual output. The Gradient Descent Update Rule $wnew=wold-\alpha\nabla wLwnew=wold-\alpha\nabla wL$ is a primary example of how weights are iteratively adjusted in the direction that most reduces the loss.

Activation Functions

Activation functions introduce non-linear properties to the network which allows the model to learn more complex patterns. Two common examples are:

3. **Sigmoid Activation Function**: $\sigma(z)=11+e-z$ is particularly useful in binary classification as it maps any real-valued number into the (0,1)(0,1) interval, representing a probability-like output.
4. **Softmax Function for Multi-Class Classification**: For scenarios where classifications are mutually exclusive, the SoftMax function normalizes the outputs, converting them into probability scores summing to one.

Loss Functions

To quantify how well a network performs, loss functions are crucial. They provide a metric to evaluate the errors made by a network during training.

5. **Cross-Entropy Loss**: In classification problems, cross-entropy helps measure the performance by calculating the total entropy between the predicted probabilities and the actual distribution.

Regularization Techniques

Overfitting is a common problem where a model learns the detail and noise in the training data to an extent that it negatively impacts the performance of the model on new data.

6. **L2 Regularization**: Also known as Ridge Regression, it adds a penalty on the size of coefficients. $R(w)$=$\lambda\|w\|2R(w)$=$\lambda\|w\|2$ where $\lambda\lambda$ is a complexity parameter that controls the amount of shrinkage: the larger the value of $\lambda\lambda$, the greater the amount of shrinkage.
7. **Dropout**: A different type of regularization where randomly selected neurons are ignored during training, reducing the chance for overfitting.

Advanced Optimization

Beyond simple gradient descent, more sophisticated algorithms can improve convergence.

8. **Stochastic Gradient Descent (SGD)** and **Adam Optimizer**: While SGD updates parameters with a subset of training data rather than the full dataset for faster computations, Adam, on the other hand, computes adaptive learning rates for each parameter.

Beyond Classical Layers: Specialized Operations

Deep learning also involves more complex operations which are suited for specific data types and problems.

9. **Backpropagation Through Time in RNNs**: Essential for training Recurrent Neural Networks, it involves moving backward through the network to calculate gradients over time, crucial for sequence prediction problems.
10. **Convolution Operations in CNNs**: Used predominantly in processing image data, this operation extracts features from local image patches through a filter.

Decision Boundary Equation:

$$f(x) = w^T x + b - 0 \qquad (1)$$

This equation defines the decision boundary of a linear classifier, where w is the weight vector, x is the input feature vector, and b is the bias. The decision boundary separates different classes in the feature space.

Gradient Descent Update Rule:

$$w_{\text{new}} = w_{\text{old}} - \alpha \nabla_w L \qquad (2)$$

This equation represents the update rule for the weights w in gradient descent optimization, where α is the learning rate and $\nabla_w L$ is the gradient of the loss function L with respect to the weights.

Sigmoid Activation Function:

$$\sigma(z) = \frac{1}{1+e^{-z}} \qquad (3)$$

The sigmoid function is used as an activation function in neural networks to map real valued inputs to the $(0,1)$ range, facilitating binary classification tasks.

SoftMax Function for Multi-class Classification:

$$\text{Softmax}\left(z_i\right)=\frac{e^{2^2}}{\sum_i e^3} \tag{4}$$

SoftMax function is used in multi-class classification to convert the output logits z into probabilities by normalizing them across all classes.
Cross-Entropy Loss for Binary Classification:

$$L(y, \hat{y})=-[y \log (\hat{y})+(1-y) \log (1-\hat{y})] \tag{5}$$

Cross-entropy loss measures the performance of a classification model whose output is a probability value between 0 and 1 . It penalizes the deviation from actual labels y and predicted probabilities $\hat{y}$.

L2 Regularization Term:

$$R(w)=\lambda \| w \|^2 \tag{6}$$

L2 regularization is used to prevent overfitting by adding a penalty term $R(w)$ to the loss function, which is proportional to the square of the magnitude of the weights, controlled by the regularization parameter λ.

Stochastic Gradient Descent (SGD):

$$w=w-\alpha \nabla_w L_i \tag{7}$$

Stochastic Gradient Descent updates the model weights using the gradient of the loss function L_i with respect to a randomly selected subset of data, improving optimization speed.

Backpropagation Through Time (For RNNs):

$$\frac{\partial L}{\partial w}=\sum_t \frac{\partial L_u}{\partial w} \tag{8}$$

This equation is used to calculate the gradients of loss function L with respect to weights w in recurrent neural networks (RNNs), summed over all time steps t.
Convolution Operation in CNNs:

$$(f * g)(t)=\int \ f(\tau) g(t-\tau) d \tau \tag{9}$$

In the context of convolutional neural networks (CNNs), this equation describes the convolution operation between an input function f and a kernel g, used for feature extraction.
Max Pooling in CNNs:

$$\text{MaxPooling}\ (A)=\max_{i, j \in A} a_{i j} \tag{10}$$

Max pooling reduces the dimensionality of each feature map in CNNs by taking the maximum value over a specified window A.

This selection includes equations relevant to various aspects of deep learning, from basic operations to more complex functions involving optimization and regularization.

Euler's Formula for Complex Exponentials:

$$e^{i\theta} = \cos(\theta) + i\sin(\theta) \qquad (11)$$

This fundamental equation in complex analysis is used for transforming complex numbers into their exponential form, which can be useful in signal processing aspects of neural networks. Laplacian Operator in Graph Theory:

$$L = D - A \qquad (12)$$

The Laplacian matrix L of a graph is used in graph neural networks, where D is the degree matrix and A is the adjacency matrix, helping to define the structure of the graph. Eigenvalue Decomposition for PCA:

$$X^T X v = \lambda v \qquad (13)$$

Principal Component Analysis (PCA) involves finding the eigenvalues λ and eigenvectors v of the covariance matrix $X^T X$ to reduce dimensions and extract features. ReLU Activation Function:

$$\text{ReLU}\,(x) - \max(0, x)$$

The Rectified Linear Unit (ReLU) activation function is widely used in deep learning for introducing non-linearity, improving training speeds and convergence. Adam Optimizer Update Rule:

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{6_t + e}} \hat{m}_t \qquad (14)$$

Adam optimization algorithm uses adaptive learning rates for each parameter by estimating first (m) and second (v) moments of the gradients, improving the efficiency of network training.

Batch Normalization Transform:

$$y = \gamma \left(\frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} \right) + \beta \qquad (15)$$

Batch normalization stabilizes and accelerates deep network training by normalizing the inputs (layer activations) x using mean μ and variance σ^2, scaled by γ and shifted by β. Hinge Loss for SVM:

$$L = \max(0, 1 - y \cdot f(x)) \qquad (16)$$

Used primarily in support vector machines (SVMs) for classification, hinge loss penalizes misclassifications and instances where the classification margin is not maintained. Entropy in Information Theory:

$$H(X) = -\sum_i p(x_i) \log p(x_i) \qquad (17)$$

Entropy measures the uncertainty or impurity in a dataset and is used in various machine learning algorithms to gauge information gain and decision making. Gaussian Kernel in SVMs:

$$K(x,x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) \quad (18)$$

The Gaussian or RBF kernel is used in SVMs for transforming the input space into a higher-dimensional space, allowing for the handling of nonlinear boundaries. Dropout Regularization:

$$\begin{aligned} &r_j \sim \text{Bernoulli } (p) \\ &\bar{x} = x * r \end{aligned} \quad (19)$$

Dropout prevents overfitting in neural networks by randomly setting a fraction p of input units to zero at each update during training.

Attention Mechanism in Transformers:

$$\text{Attention } (Q,K,V) = \text{softmax}\left(\frac{QK^T}{Vd_k}\right)V \quad (20)$$

In the context of transformer models, this equation calculates the attention weights and outputs by scaling the dot products of queries Q and keys K with values V. Gated Recurrent Unit (GRU) Update Gate:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (21)$$

The update gate in GRU helps the model to determine how much of the past information (from previous time steps) needs to be passed along to the future.
Gated Recurrent Unit (GRU) Reset Gate:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (22)$$

The reset gate in GRU controls how much of the past information to forget, which helps in capturing dependencies and managing vanishing gradient issues in sequences. Gated Recurrent Unit (GRU) Current Memory Content:

$$\bar{h}_t = \tanh\,(W \cdot [r_t * h_{t-1}, x_t]) \quad (23)$$

This represents the candidate activation or new memory content in GRU, combining the reset gate's output with the current input.

Gated Recurrent Unit (GRU) Final Memory at Current Step:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \bar{h}_t \quad (25)$$

This final state for GRU at time t blends the old state h_{t-1} and the new candidate state $\bar{h}_t$ based on the update gate z_t.

L-BFGS Update Formula for Optimization:

$$s_k = x_{k+1} - x_k \quad (26)$$
$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k) \quad (27)$$
$$\rho_k = \frac{1}{y_k^T s_k} \quad (28)$$
$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \quad (29)$$

L-BFGS is a quasi-Newton method that approximates the Hessian matrix involved in the optimization of nonlinear functions, suitable for large-scale problems in deep learning.

Tikhonov Regularization (Ridge Regression):

$$\min_{w} \| Xw - y \|^2 + \lambda \| w \|^2 \qquad (30)$$

Tikhonov regularization, or ridge regression, is used to solve regression problems by imposing a penalty on the size of coefficients.

Elastic Net Regularization:

$$\min_{w} \| Xw - y \|^2 + \lambda_1 \| w \|_1 + \lambda_2 \| w \|^2 \qquad (31)$$

Elastic net combines L1 and L2 regularization, helping in variable selection (L1) and shrinking coefficients (L2), beneficial in scenarios with highly correlated variables.
Kullback-Leibler Divergence for Probability Distributions:

$$D_{KL}(P \| Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \qquad (32)$$

Kullback-Leibler divergence measures how one probability distribution diverges from a second, expected probability distribution.
Dynamic Programming Recurrence Relation:

$$V(i, w) = \max(V(i-1, w), V(i-1, w-w_i) + v_i) \qquad (33)$$

This equation is used in the knapsack problem, a common problem in combinatorial optimization, where $V(i, w)$ is the maximum value that can be achieved with the first i items and a weight limit w.

These equations collectively illustrate the mathematical and computational complexity involved in designing and optimizing deep learning models, particularly when dealing with sophisticated classifiers and large datasets. The equations and concepts discussed represent the backbone of deep learning technologies. By integrating these mathematical principles, deep learning models can be optimized and tailored to specific problems more effectively. As we continue to push the boundaries of what is possible with these models, understanding and applying these foundational equations remain critical.Each component, from linear transformations and regularization to loss functions and specialized neural network operations, plays a crucial role in building robust, efficient, and scalable models. The continuous evolution of these mathematical frameworks is what will drive future innovations in deep learning, enabling smarter and more intuitive artificial intelligence systems. Whether for simple applications or complex, domain-specific tasks, the integration of complex mathematical operators and advanced optimization techniques will remain at the forefront of technological advancements in AI.

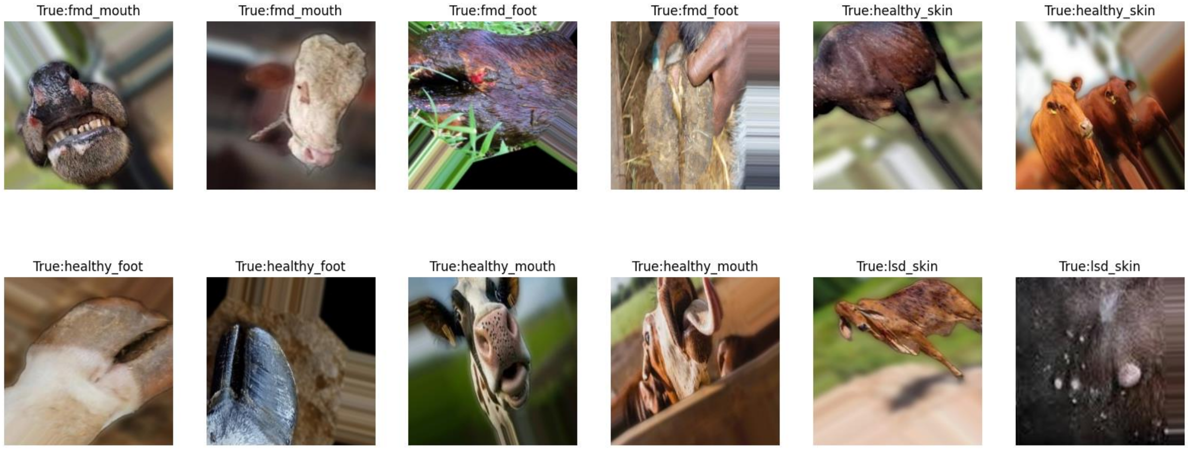

Figure 1. Different Samples of Foot-Mouth Disease and Skin Disease

The introduction of these technologies underscores a larger trend toward the automation and enhancement of disease detection methods in agriculture. Deep learning, through the use of convolutional neural networks (CNNs), enables the analysis of cattle images to identify and classify diseases accurately. This not only aids in early detection but also enhances the overall management of cattle health, contributing to more sustainable agricultural practices. Our research aims to further these advancements by focusing on the development of an efficient cattle disease detection system that is accessible and user-friendly. By leveraging deep learning and smartphone technology, our proposed system aims to empower farmers and veterinarians to diagnose diseases accurately and efficiently, even in resource-constrained environments. This approach addresses the critical problem of delayed or inaccurate disease diagnosis, which can lead to prolonged animal suffering, heightened economic losses, and increased risk of disease spread. In essence, the problem our research seeks to solve is the significant gap in timely and accurate disease detection in cattle, which can be transformative for cattle farming. By employing advanced AI technologies and making these tools accessible through smartphones, we can revolutionize how diseases are managed in the livestock industry. The following sections of this paper will detail the methodology, implementation, evaluation results, and potential impact of our system, aiming to provide a comprehensive solution to a pressing issue in global agriculture.

2. RELATED STUDY

Cattle farming, a cornerstone of global agriculture, faces significant challenges due to diseases that can severely impact livestock health and productivity. This literature review explores the recent advancements in machine learning and deep learning techniques applied to cattle disease detection, classification, and management. It draws on various studies to highlight the current technologies, methodologies, and the effectiveness of these approaches, as well as the existing gaps and future directions in research.

Table 1. Analysis of Methodologies

Reference	Research Purpose	Methodologies Used	Key Findings	Challenges/Research Gaps
Y. Li et al.	Classification of cattle behaviors	KNN, Random Forest, Extreme Boosting	High accuracy in behavior classification	Limited temporal information affects accuracy
K. Liakos et al.	Real-time lameness detection	ANN, SVM, Random Forest	Effective in lameness detection	Needs more reliable and practical systems
D. Wu et al.	Detection of cattle lameness	YOLOv3, SVM, KNN, DTC	High accuracy in lameness detection	Limited datasets
M. E. Pastell et al.	Lameness detection using PNN	Probabilistic Neural Network (PNN), 4-balance system data	High accuracy in identifying lameness	System may not be suitable for all farms
N. V. Kishan et al.	Health monitoring and disease prediction	Data mining, k-NN, Naive Bayes, SVM, IoT	Focus on rare diseases and veterinary care	Integration with existing practices
G. Rai et al.	Prediction of Lumpy Skin Disease	CNN, ANN, VGG-16, VGG-19, Inception-v3	High accuracy in disease prediction	No standard dataset available
Y. Qiao et al.	Multi-cattle segmentation	Mask R-CNN	High accuracy in segmentation	Superior to traditional segmentation methods
R. Dulal et al.	Cattle identification using YOLOv5	YOLOv5	Effective in overcoming RFID limitations	Cost and scalability analysis needed

Y. Li et al. developed a method for classifying multiple cattle behaviors using machine learning algorithms like K-nearest neighbors and random forests, achieving a high F1 score accuracy. However, their study noted limitations in capturing temporal information which affected the accuracy of movement classifications such as feed tossing and rolling biting . Several researchers have focused on detecting lameness, a prevalent issue in cattle which can significantly affect their welfare and farm productivity. K. Liakos et al. trained Artificial Neural Networks (ANN) and used Support Vector Machines (SVM) for real-time lameness detection. This approach highlighted the potential of machine learning to improve the accuracy and reliability of automated lameness detection systems . D. Wu et al. also targeted lameness detection using a combination of YOLOv3, SVM, K-Nearest Neighbor, and Decision Tree Classifiers. Their approach achieved a notable accuracy of 98.57%, demonstrating the efficacy of deep learning algorithms in diagnosing complex conditions like lameness .M. E. Pastell et al. utilized a Probabilistic Neural Network (PNN) model using data from a 4-balance system to detect lameness with high accuracy. This system, while effective, was noted to be potentially unsuitable for all farm types due to its design and implementation challenges .N. V. Kishan et al. explored the use of data mining techniques alongside IoT-based systems for cattle health monitoring and disease prediction. Their approach used classification methods such as k-nearest neighbors, Naive Bayes, and SVM to focus on rare diseases and improve veterinary practices. G. Rai et al. applied convolutional neural networks (CNN) like VGG-16 and

Inception-v3 for predicting Lumpy Skin Disease, achieving a high accuracy of 92.5%. However, the absence of a standard dataset posed a significant challenge . Y. Qiao et al. introduced a method for multi-cattle segmentation using Mask R-CNN, achieving a mean pixel accuracy of 92%. This method surpassed traditional instance segmentation methods like SharpMask and DeepMask, highlighting the potential of deep learning in precise livestock management . R. Dulal et al. used the YOLOv5 model for cattle identification, focusing on overcoming the limitations of RFID systems through deep learning techniques. This study provided insights into cost-effective and scalable solutions for cattle management. The integration of advanced technologies in real-world farming operations remains a significant challenge. The need for systems that are not only accurate but also user-friendly and adaptable to different farming conditions is critical for widespread adoption [10-19].

3. MATERIALS AND METHODS

This research focuses on creating an advanced system for detecting cattle diseases using deep learning techniques, particularly Convolutional Neural Networks (CNNs) and transfer learning. The primary goal is to assist farmers and veterinarians in the early detection and diagnosis of cattle diseases, thereby enhancing animal health and productivity. The methodology utilizes state-of-the-art deep learning models, known for their effectiveness in image classification, tailored specifically for identifying various cattle diseases from visual cues[20].

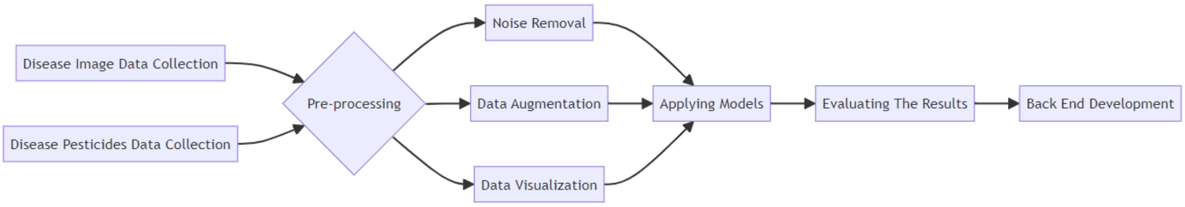

Figure 2. Process Flow Diagram

Figure 2 displays the workflow from data acquisition to the output of results, outlining the steps involved in processing and analyzing the images. **Figure 3** provide insights into the distribution of images across different classes, highlighting the diversity and balance of the dataset which is crucial for training unbiased models.**Figure 4** shows a pie chart that represents the proportion of images per disease category, emphasizing the dataset's variety and the need for a model capable of recognizing a broad spectrum of disease indicators.

Dataset Visualization and Acquisition

The foundation of any deep learning application lies in the quality and structure of the dataset used. For this study, a comprehensive dataset consisting of 455 images was compiled. These images include various cattle diseases and healthy specimens, categorized into different classes based on the disease type. Each class is distinctly labeled to facilitate supervised learning, where the model learns to associate specific images with their corresponding disease labels.

Data Pre-processing

Data pre-processing is critical to enhance the data quality, which in turn, improves the learning efficiency and performance of the model.

Data Noise Removal: Noise reduction is pivotal in image processing, especially for medical and biological imaging where clarity and detail are essential for accurate diagnosis. Techniques were implemented to reduce noise and remove irrelevant background elements, enhancing the contrast and focus on key features essential for disease classification[22].

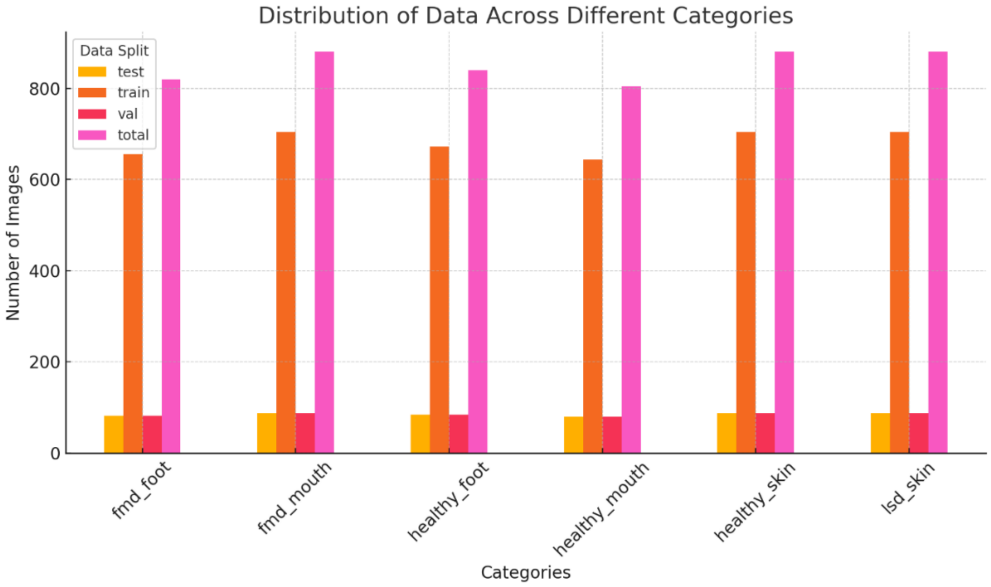

Figure 3. Sample Distribution of Database

Data Augmentation: To bolster the model's robustness and its ability to generalize across different visual representations of cattle diseases, various data augmentation techniques were applied. These included geometric transformations like rotations, flips, and translations; color adjustments to mimic different lighting conditions; and zooms and crops to vary the image focus and perspective. Additionally, artificial noise and blur were introduced to ensure the model's effectiveness under varied photographic conditions, simulating real-world scenarios where images may not always be perfect.

Deep Learning Models

The project utilizes several advanced deep learning models, adapting their architecture through the process of transfer learning to meet the specific requirements of cattle disease detection.

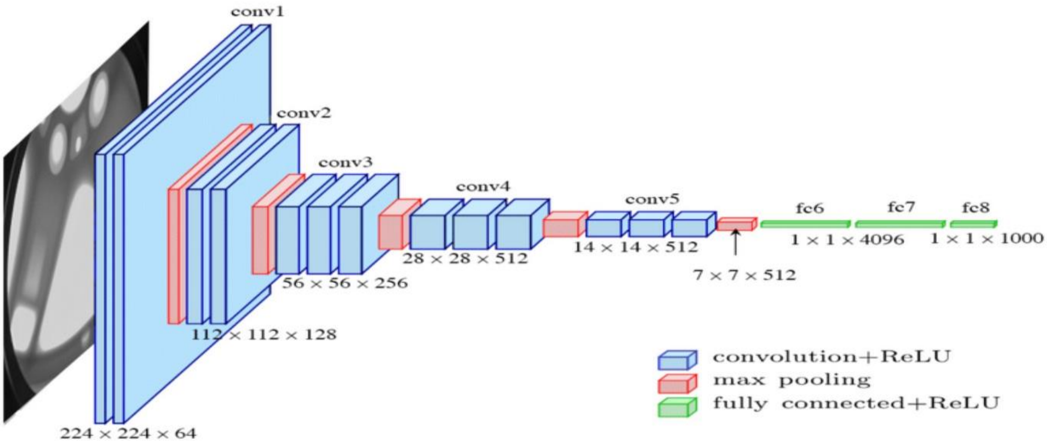

Figure 4. Design of VGG 16 Based Classifier

VGG16: VGG16, known for its depth and simplicity, was selected for its proven track record in image classification tasks. The model, pre-trained on the ImageNet database, was fine-tuned on the cattle disease dataset. The last few layers were specifically modified to align with the number of disease classes identified in the dataset. This fine-tuning allows the pre-trained network to adapt its learned features to new, specific tasks of recognizing cattle diseases[23-25].

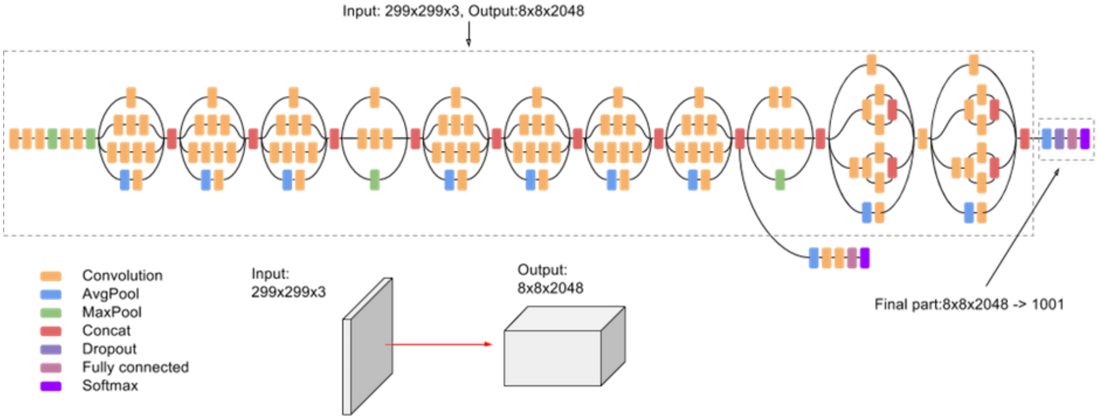

Figure 5. Design of Inception V3 Based Classifier

Inception V3: Inception V3 is another model chosen for its sophisticated architecture designed to capture information at multiple scales using various convolutions and pooling strategies. The model was fine-tuned with additional layers like batch normalization and dropout to enhance training stability and minimize overfitting, adapting it to the nuances of cattle disease imaging.

ResNet50: ResNet50 employs residual learning to facilitate the training of very deep networks by incorporating skip connections that jump over some layers. These connections help mitigate the vanishing gradient problem, allowing for deeper network architectures without a loss in performance. ResNet50 was tailored for deeper and more precise feature extraction, offering substantial computational efficiency over other deep networks without residual connections [27].

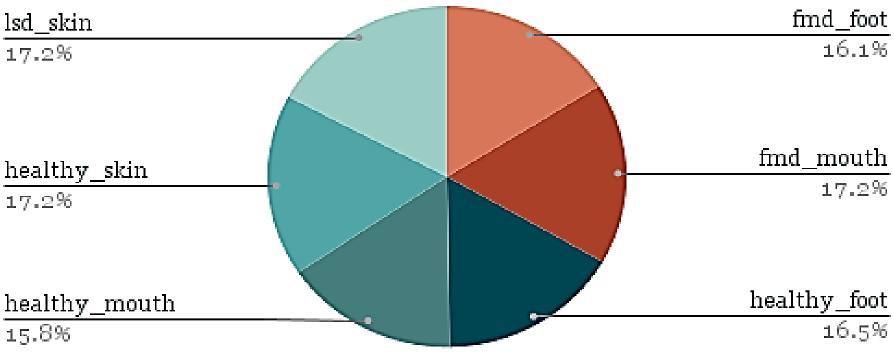

Figure 6. Distribution of Diseases in Database

Training Details:

- **Optimization Algorithms:** The Adam optimizer was used due to its adaptive learning rate features, enhancing the convergence speed.
- **Loss Functions:** Cross-entropy loss, a standard for multiclass classification tasks, was employed to compute the model's loss.
- **Performance Metrics:** Metrics such as Accuracy, Precision, Recall, and F1-Score were meticulously recorded to evaluate the model's performance and ensure reliability.

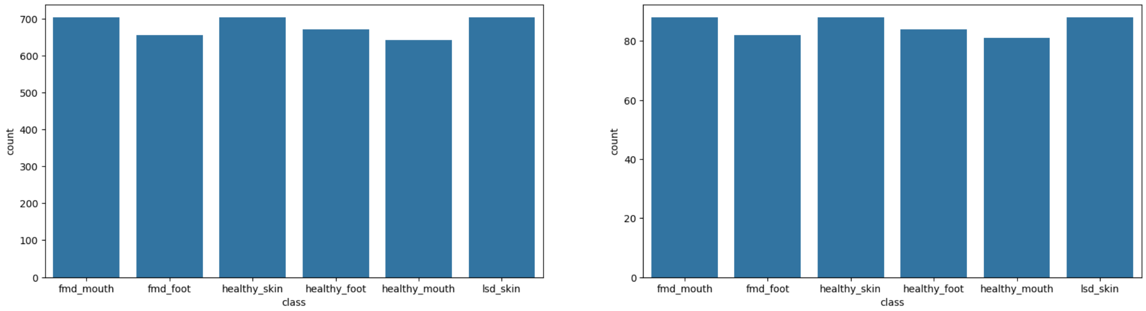

Figure 7. Analysis of Distribution of Diseases in Database

Evaluation:

- **Validation Strategy:** The dataset was strategically divided into training, validation, and test sets, ensuring that the model's performance is rigorously evaluated on unseen data.
- **Cross-validation:** Used to ascertain the consistency and reliability of the model across different data subsets, ensuring that the model performs well in various scenarios and setups.

This research methodology leverages the powerful capabilities of CNNs and transfer learning to develop a robust, accurate, and efficient system for cattle disease detection. By combining sophisticated image processing techniques with cutting-edge deep learning technologies, this system is poised to make significant contributions to veterinary medicine. It promises to enhance the capabilities for early disease detection and management in cattle, thereby improving animal health, increasing farm productivity, and reducing economic losses associated with cattle diseases [28-32].

4. RESULTS & DISCUSSION

Recent advances in AI, particularly the development of deep learning technologies, have opened up new avenues for innovation in disease detection. CNNs, a class of deep neural networks, are particularly suited for image recognition tasks and have been successfully applied in various fields including medical imaging and autonomous driving. These networks automatically learn to detect complex patterns in data, making them ideal for recognizing signs of disease in images of cattle. The primary objective of this study was to develop a robust automated system capable of detecting specific cattle diseases from images using CNN architectures such as InceptionV3, MobileNetV2, and DenseNet169. This system aimed to classify images into distinct categories based on the presence of foot, mouth, and lumpy skin diseases, leveraging a dataset compiled from various online sources. A comprehensive dataset was essential for training and testing the deep learning models. The dataset consisted of images classified into different disease categories, providing a varied pool of data points for the models to learn from. The images were collected from openly available internet sources, ensuring a diverse representation of disease manifestations. This diversity is critical in training robust models that can generalize well across different real-world scenarios. The study utilized the Python programming language, known for its extensive libraries and frameworks that facilitate data handling, processing, and model development. Key technologies used included TensorFlow and Keras for building and training the CNN models, and NumPy for performing mathematical operations essential for data preprocessing and analysis.

The performance of each model was evaluated using standard metrics derived from the confusion matrix, including accuracy, precision, recall, and the F1-score. These metrics provide a comprehensive understanding of model performance, highlighting strengths and weaknesses in specific areas such as the model’s ability to correctly identify disease-positive cases (precision) and its sensitivity in detecting these cases among all positive instances (recall).

Classification Accuracy: The accuracy of classification is determined by determining the percentage of data points that have been successfully classified out of the total number of data points. Calculated by taking the total number of data points and dividing it by the sum of true positives (TP) and true negatives (TN), it includes the following:

$$\textbf{Accuracy}=\frac{\mathbf{TP+TN+FP+FN}}{\mathbf{TP+TN}} \tag{34}$$

Precision: Accuracy can be defined as the ratio of the actual positive to the total number of positives anticipated. A model with a high precision has a low false positive rate, which means that it only sometimes incorrectly identifies negative occurrences as positive. Conversely, a low precision suggests that the model tends to make a significant number of false positive predictions, which can be problematic in scenarios where false positives are costly or undesirable as in fraud detection, and spam filtering.

$$\textbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}} \qquad (35)$$

Recall, - In classification tasks, a performance indicator that is also known as sensitivity or true positive rate is utilized to evaluate the capability of a model to accurately identify all positive cases from the entire number of actual positive examples that are contained inside the dataset. A formula that is used to compute it is as follows:

$$\textbf{Recall-=}\frac{\mathbf{TP}}{\mathbf{TP+FN}} \qquad (36)$$

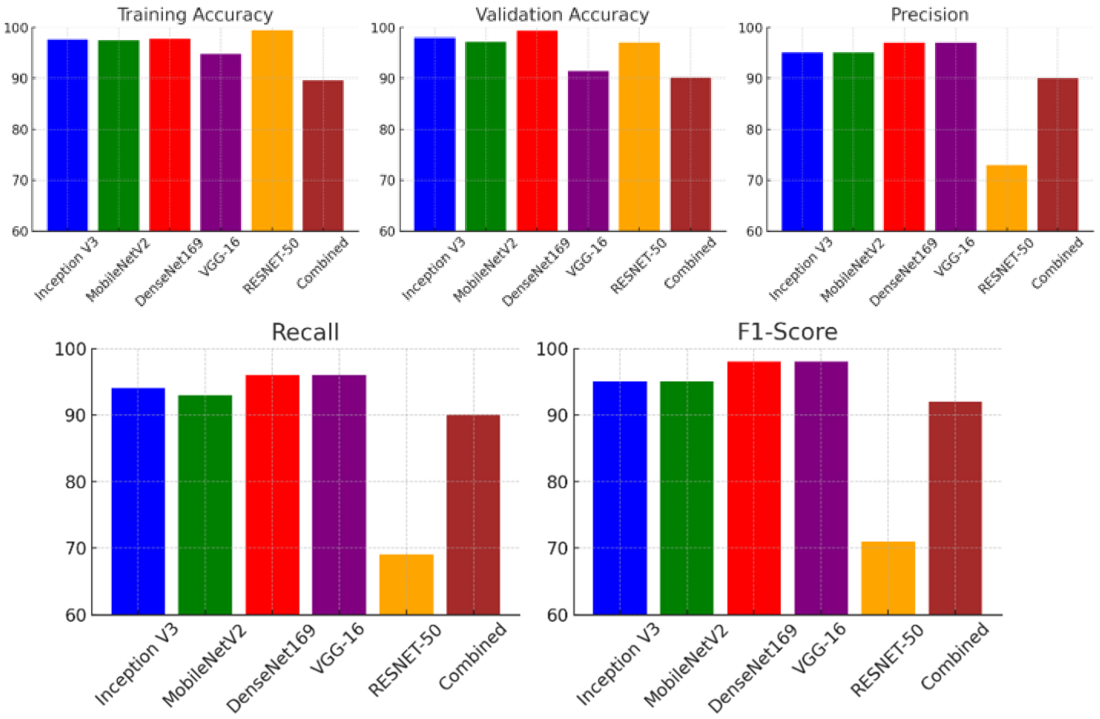

Figure 8. Performance Parameters of Classifier

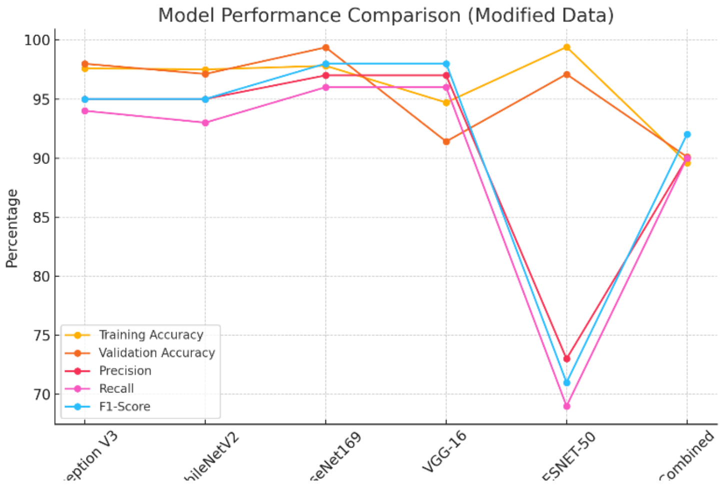

Figure 9. Comparative Analysis of Performance Parameters of Classifier

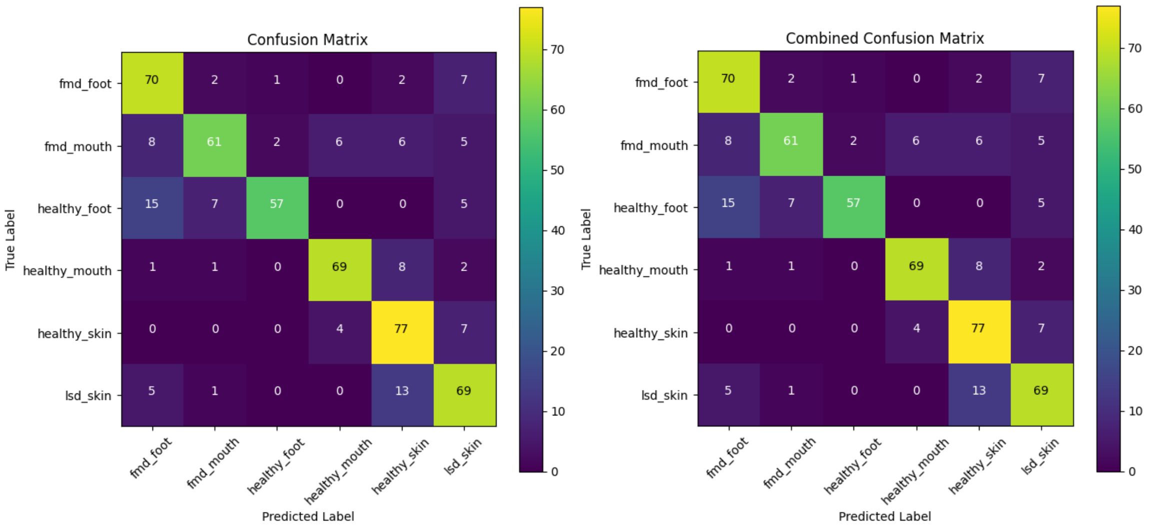

Figure 10. Comparative Analysis of Confusion Matrix for VGG-16 and Combined Classifier

In the contemporary landscape of agricultural technology, the implementation of artificial intelligence (AI), particularly deep learning models, offers a promising frontier for enhancing livestock disease detection. This analysis focuses on evaluating the performance of various convolutional neural networks (CNNs) employed in the detection of cattle diseases, including Inception V3, MobileNetV2, DenseNet169, VGG-16, and RESNET-50. The study uses a tailored dataset that reflects common cattle ailments, such as foot, mouth, and lumpy skin diseases. The dataset comprises images sourced from various online repositories, classified into specific disease categories to train the CNN models. This approach is designed to simulate a real-world application where visual data from livestock would be analyzed to predict disease presence. The models were evaluated based on several metrics, including training and validation accuracy, precision, recall, and F1-score, which provide a comprehensive view of model performance.The results presented in the study are as follows:

- **Inception V3**: Demonstrated high reliability with a training accuracy of 97.60% and validation accuracy of 97.98%. The model showed robust precision and recall, making it suitable for scenarios where both false positives and false negatives carry significant consequences.
- **MobileNetV2**: Featured slightly lower validation accuracy than Inception V3, indicating potential overfitting or sensitivity to the dataset's variability. Nevertheless, its high precision and recall suggest it remains an effective tool for disease detection.
- **DenseNet169**: Emerged as the top performer with the highest validation accuracy (99.37%) and excellent precision and recall. Its architecture possibly offers better feature extraction capabilities, which is crucial for detailed image analysis in disease identification.
- **VGG-16**: While it had the lowest validation accuracy among the top performers, its precision and recall were high. This model might be more sensitive to the training dataset's nuances and could benefit from further tuning.

- **RESNET-50**: Although it had the highest training accuracy (99.40%), its lower validation accuracy and significantly lower precision and recall indicate issues with generalizability and potential overfitting.
- **Combined**: This model, likely an ensemble of different architectures, showed balanced performance across all metrics, demonstrating the utility of model ensembling to stabilize detection performance.

Comparing these results with historical data from other studies underscores the advancement in deep learning applications for agricultural uses. The DenseNet169 model, in particular, stands out by setting a high benchmark in validation accuracy and F1-score, indicating superior generalization over unseen data, a critical factor for practical deployment.

The high performance of these CNN models suggests they could be integrated into mobile applications for real-time disease detection, providing farmers with timely, accurate diagnostics, thus allowing for quicker response and treatment. Future research could explore incorporating additional disease categories, enhancing data diversity, and deploying these models in field trials to further validate their effectiveness in practical setting.

5. CONCLUSION

In conclusion, the application of CNNs in detecting cattle diseases has demonstrated significant potential to revolutionize disease management in livestock. By continuing to refine these models and expand their capabilities, the agricultural sector can look forward to greater efficiency and accuracy in livestock healthcare, ultimately leading to improved productivity and reduced losses due to diseases. This study not only contributes to the field of agricultural AI but also sets the stage for future innovations that could further enhance the capabilities of AI in managing livestock health. The potential of AI to impact livestock disease management positively is immense and largely untapped. As demonstrated by the success of CNNs in this study, there is a clear path forward that involves technological innovation, careful consideration of ethical and practical concerns, and close collaboration between diverse stakeholders. By continuing to pursue these avenues, we can look forward to a future where livestock management is more data-driven, efficient, and responsive to the challenges posed by diseases. This will not only improve the health and productivity of livestock but also the economic viability of farms around the globe, securing food supply chains and enhancing the sustainability of agricultural practice.

REFERENCES

[1] G. Rai, Naveen, A. Hussain, A. Kumar, A. Ansari, and N. Khanduja, “A deep learning approach to detect lumpy skin disease in cows,” in *Computer Networks, Big Data and IoT: Proceedings of ICCBI 2020*. Springer, 2021, pp. 369–377.

[2] M. Pastell and M. Kujala, “A probabilistic neural network model for lameness detection,” *Journal of dairy science*, vol. 90, no. 5, pp. 2283–2292, 2007.

[3] G. Li, G. E. Erickson, and Y. Xiong, “Individual beef cattle identification using muzzle images and deep learning techniques,” *Animals*, vol. 12, no. 11, p. 1453, 2022.

[4] N. Kreˇsi´c, I. Sˇimi´c, T. Bedekovi´c, Zˇ. Acinger-Rogi´c, and I. Lojki´c, “Evaluation of serological tests for detection of antibodies against lumpy skin disease virus,” *Journal of Clinical Microbiology*, vol. 58, no.

9, pp. 10–1128, 2020.

[5] B. Lake, F. Getahun, and F. T. Teshome, “Application of artificial intelligence algorithm in image processing for cattle disease diagnosis,” *Journal of Intelligent Learning Systems and Applications*, vol. 14, no. 04, pp. 71–88, 2022.

[6] B. Jiang, Q. Wu, X. Yin, D. Wu, H. Song, and D. He, “Flyolov3 deep learning for key parts of dairy cow body detection,” *Computers and Electronics in Agriculture*, vol. 166, p. 104982, 2019.

[7] V. R. Allugunti, “A machine learning model for skin disease classification using convolution neural network,” *International Journal of Computing, Programming and Database Management*, vol. 3, no. 1, pp. 141–147, 2022.

[8] A. Poursaberi, C. Bahr, A. Pluk, A. Van Nuffel, and D. Berckmans, “Real-time automatic lameness detection based on back posture extraction in dairy cattle: Shape analysis of cow with image processing techniques,” *Computers and electronics in agriculture*, vol. 74, no. 1, pp. 110–119, 2010.

[9] E. Afshari Safavi, “Assessing machine learning techniques in forecasting lumpy skin disease occurrence based on meteorological and geospatial features,” *Tropical Animal Health and Production*, vol. 54, no. 1, p. 55, 2022.

[10] E. Arazo, R. Aly, and K. McGuinness, “Segmentation enhanced lameness detection in dairy cows from rgb and depth video,” *arXiv preprint arXiv:2206.04449*, 2022.

[11] S. Shahinfar, D. Page, J. Guenther, V. Cabrera, P. Fricke, and K. Weigel, “Predic- tion of insemination outcomes in holstein dairy cattle using alternative machine learning algorithms,” *Journal of dairy science*, vol. 97, no. 2, pp. 731–742, 2014.

[12] R. Chandrarathna, T. Weerasinghe, N. Madhuranga, T. Thennakoon, A. Gamage, and E. Lakmali, “’the taurus’: Cattle breeds & diseases identification mobile application using machine learning,” *arXiv preprint arXiv:2302.10920*, 2023.

[13] Y. Qiao, D. Su, H. Kong, S. Sukkarieh, S. Lomax, and C. Clark, “Individual cattle identification using a deep learning based framework,” *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 318–323, 2019.

[14] M. Williams, N. Mac Parthal´ain, P. Brewer, W. James, and M. Rose, “A novel behavioral model of the pasture-based dairy cow from gps data using data mining and machine learning techniques,” *Journal of dairy science*, vol. 99, no. 3, pp. 2063–2075, 2016.

[15] Y. Li, H. Shu, J. Bindelle, B. Xu, W. Zhang, Z. Jin, L. Guo, and W. Wang, “Classification and analysis of multiple cattle unitary behaviors and movements based on machine learning methods,” *Animals*, vol. 12, no. 9, p. 1060, 2022.

[16] Y. Qiao, M. Truman, and S. Sukkarieh, “Cattle segmentation and contour extraction based on mask r-cnn for precision livestock farming,” *Computers and Electronics in Agriculture*, vol. 165, p. 104958, 2019.

[17] K. Liakos, S. P. Moustakidis, G. Tsiotra, T. Bartzanas, D. Bochtis, and C. Parisses, “Machine learning based computational analysis method for cattle lameness predic- tion.” *HAICTA*, vol. 128, p. 139, 2017.

[18] D. Wu, Q. Wu, X. Yin, B. Jiang, H. Wang, D. He, and H. Song, “Lameness detection of dairy cows based on the yolov3 deep learning algorithm and a relative step size characteristic vector,” *Biosystems Engineering*, vol. 189, pp. 150–163, 2020.

[19] S. Grampurohit and C. Sagarnal, “Disease prediction using machine learning algorithms,” in *2020 International Conference for Emerging Technology (INCET)*. IEEE, 2020, pp. 1–7.

[20] N. V. Kishan, Y. S. Trinath, S. Kavalur, V. Sangamesha, and S. Reddy, “Cattle disease identification using prediction techniques,” *IJARIIE-ISSN (O)*, pp. 2395– 4396, 2021.

[21] R. Dulal, L. Zheng, M. A. Kabir, S. McGrath, J. Medway, D. Swain, and W. Swain, “Automatic cattle identification using yolov5 and mosaic augmentation: A com- parative analysis,” in *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2022, pp. 1–8.

[22] N. Abdul Ghafoor and B. Sitkowska, “A machine learning application to predict risk of mastitis in cattle

from ams sensor data," *AgriEngineering*, vol. 3, no. 3, pp. 575–583, 2021.

[23] I. Ramirez-Morales, L. Aguilar, E. Fernandez-Blanco, D. Rivero, J. Perez, and A. Pazos, "Detection of bovine mastitis in raw milk, using a low-cost nir spectrom- eter and k-nn algorithm," *Applied Sciences*, vol. 11, no. 22, p. 10751, 2021.

[24] D. M. Amin, G. Shehab, R. Emran, R. T. Hassanien, G. N. Alagmy, N. M. Hagag, M. I. Abd-El-Moniem, A. R. Habashi, E. M. Ibraheem, and M. A. Shahein, "Diagnosis of naturally occurring lumpy skin disease virus infection in cattle using virological, molecular, and immunohistopathological assays," *Veterinary World*, vol. 14, no. 8, p. 2230, 2021.

[25] G. Limon, A. A. Gamawa, A. I. Ahmed, N. A. Lyons, and P. M. Beard, "Epidemi- ological characteristics and economic impact of lumpy skin disease, sheeppox and goatpox among subsistence farmers in northeast nigeria," *Frontiers in veterinary science*, vol. 7, p. 8, 2020.

[26] L. Aerts, A. Haegeman, I. De Leeuw, W. Philips, W. Van Campe, I. Behaeghel, L. Mostin, and K. De Clercq, "Detection of clinical and subclinical lumpy skin disease using ear notch testing and skin biopsies," *Microorganisms*, vol. 9, no. 10, p. 2171, 2021.

[27] V. Punyapornwithaya, O. Arjkumpa, N. Buamithup, C. Jainonthee, R. Salvador, and K. Jampachaisri, "The impact of mass vaccination policy and control measures on lumpy skin disease cases in thailand: insights from a bayesian structural time series analysis," *Frontiers in Veterinary Science*, vol. 10, p. 1301546, 2024.

[28] N. Murti, E. Safi P. Srianto, S. Madyawati, and A. Rofikoh, "Prevalence and progression of lumpy skin disease cases in cattle over the six months leading up to eid al-adha in 2023 in the cirebon district of west java province, indonesia," in *IOP Conference Series: Earth and Environmental Science*, vol. 1292, no. 1. IOP Publishing, 2024, p. 012037.

[29] G. Mackereth, K. Rayner, A. Larkins, D. Morrell, E. Pierce, and P. Letchford, "Surveillance for lumpy skin disease and foot and mouth disease in the kimberley, western australia," *Australian Veterinary Journal*, 2024.

[30] A. Elsonbaty, M. Alharbi, A. El-Mesady, and W. Adel, "Dynamical analysis of a novel discrete fractional lumpy skin disease model," *Partial Differential Equations in Applied Mathematics*, vol. 9, p. 100604, 2024.

[31] Alaria, S. K. "A.. Raj, V. Sharma, and V. Kumar."Simulation and Analysis of Hand Gesture Recognition for Indian Sign Language Using CNN"." *International Journal on Recent and Innovation Trends in Computing and Communication* 10, no. 4 (2022): 10-14.

[32] Ashish Raj, Vijay Kumar, S. K. A. V. S. Design Simulation and Assessment of Prediction of Mortality in Intensive Care Unit Using Intelligent Algorithms. *MSEA* 2022, *71*, 355-367.