Journal of Agriculture and Environment for International Development - JAEID 2025, 119 (1): 125 - 148
DOI: 10.36253/jaeid-16397

Realtime Monitoring of Animal Behavior Using Deep Learning Models

ROMESA RAO[1, *], SALMAN QADRI [1], RAO KASHIF [2]

[1] *Institute of Computing, Muhammad Nawaz Shareef University of Agriculture, Multan, Pakistan.*

[2] *Faculty of Engineering & Computing, National University of Modern Languages, Pakistan.*

** Correspondence details: romesarao72@gmail.com*

Submitted on: 2024, 26 July; accepted on 2025, 08 January. Section: Research Papers

Abstract: Accurate monitoring of animal health and behavior is crucial for improving welfare and productivity in livestock management. Traditional observation methods are time-consuming and prone to subjective bias. To address these challenges, we propose an automated system for behavioral pattern using deep learning-based pose estimation techniques. Specifically, we utilize ResNet-50, a deep convolutional neural network, to detect key anatomical landmarks such as the nose, eyes, ears, and body center. By tracking these keypoints, we generate movement trajectories that help identify behavioral patterns. For behavior classification, we initially applied a decision tree algorithm, achieving an accuracy of 60%. To enhance performance, we implemented a random forest classifier, which significantly improved the accuracy to 96%. The system tries to classify seven key behaviors: "stand," "sit," "eat," "drink," "aggressive," "sit with legs tied," and "let go of the tail." The random forest model achieved the highest accuracy in detecting "standing" and "aggressive" behaviors, while lower accuracy was observed for "eating" behavior. Additionally, our pose estimation model demonstrated high precision and recall metrics, indicating robust performance in keypoint detection with minimal deviation from ground truth annotations. This automated system reduces the need for manual observation and provides a reliable tool for monitoring some animal behavior. The potential applications extend to various domains, including animal studies and livestock management, offering a scalable and user-friendly solution for real-time behavior analysis.

Keywords: Pose Estimation, Behavior Classifier, ResNet-50, Trajectory Analysis, Random Forest, Decision Tree

Introduction

Animals cannot communicate their discomfort or health issues verbally like we do, but they convey a great deal through their body movements and behavior. By closely monitoring these movements, we can detect abnormalities or signs of illness. In our study on real-time animal behavior monitoring, we focus on tracking postures and movements over time to identify key behavioral patterns. We can determine how often a cow sits, stands, eats, or displays aggression throughout the day. This sequence of behaviors, analyzed using time series data, provides insights into the animal's health and well-being, allowing us to quickly detect issues that may need attention. Managing herds of cattle has become more challenging due to rising costs in feed and increased labor demands. Cattle experience significant physiological changes during various stages, such as preparation for giving birth, which leads to different postures like sitting,

standing, and transitions between these states. However, cows with calving difficulties may show abnormal behaviors, which can be early indicators of health issues. Traditionally, monitoring animal behavior in cows has depended on manual observation, which is time-consuming, error-prone, and demands constant attention. This study introduces an automated system that uses advanced pose estimation techniques to track cattle behavior in real time. Our model focuses on key postures linked to health indicators such as sitting, standing, eating, or aggressive behavior and continuously monitors these actions. By automating this process, we provide a more efficient, accurate, and reliable solution for livestock management, enhancing both animal welfare and farm productivity.

The researchers employed DeepLabCut to accurately generate key posture landmarks from field data for dairy cows (Li, Kang, Zhang, Liu, & Yu, 2023). The primary goal of this study was to examine how specific cow behaviors, such as sitting, standing, and eating, relate to their overall health and well-being. By continuously monitoring these behaviors in real time, we gained valuable insights into patterns that can signal potential health issues. This approach enables early detection of abnormal behaviors, which are often indicative of underlying physiological or medical conditions. Additionally, work introduced SaLSa, a novel method for identifying and classifying individual elements of behavior (Sakata, 2023). This method combines semi-automatic labeling with Long Short-Term Memory (LSTM) networks to enhance behavioral classification, enabling more accurate tracking of complex animal behaviors. These advancements offer new ways to support animal welfare by identifying health concerns early and allowing for timely intervention. Using DeepLabCut, (Wiltshire et al., 2023) trained models for effective movement prediction and tracking in wild chimpanzees, bonobos, and other forest-dwelling animals.

A researchers group developed automated systems for monitoring swine water consumption and behavior (Kashiha et al., 2013). Using camera footage and machine learning, they tracked water intake and identified aggressive behaviors, enhancing farm management. Another group introduced a shape model for cows to study behavior on dairy farms (Guzhva, 2018). Some research (Wang, He, Zheng, Gao, & Zhao, 2018) used acceleration data and Global Positioning System (GPS) to classify cow behaviors, improving accuracy by combining datasets. Other authors (Li, Wu, Kang, Zhang, & Xuan, 2018) enhanced optical flow tracking, reducing background interference and improving accuracy. Computer vision and AI, like DeepLabCut by (Mathis et al., 2018), allow real-time video mapping and neural processing. DeepLabCut achieved high accuracy with minimal training data across various species. DeepLabCut's efficiency and adaptability for diverse experiments are highlighted by (Nath et al., 2019), while (Labuguen et al., 2019) used it for detailed monkey behavior analysis, achieving precise results. Monitoring animal behavior, especially in cows, has usually been done manually, which is both time-consuming and prone to mistakes.

To address these challenges, (Fujimori, Ishikawa, & Watanabe, 2020) applied computer vision to animal behavior analysis, significantly improving tracking and measurement accuracy. Additionally, (Pereira et al., 2022) developed Social Leap Estimates Animal Poses (SLEAP), a user-friendly, Graphics processing unit (GPU) independent tool for multi-animal tracking, which has been widely used in various research environments. DeepLabCut was evaluated for marker-less animal pose estimation, achieving high accuracy with minimal labeled data (Tien et al., 2022). A study that explored the integration of CNN Convolutional Neural Networks (CNN), a type of deep learning model designed for image analysis, with Long Short-Term Memory (LSTM) networks, which capture temporal dependencies in time-series data, for cow behavior classification, that is demonstrated by (Wu et al., 2021). This approach adapts to changing environmental conditions and captures subtle changes in animal behavior over time. The researchers (Chen, Zhu, & Norton, 2021) explored traditional and deep learning methods for behavior analysis, while (Avanzato, Beritelli, & Puglisi, 2022) applied You Only Look Once (YOLOv5), a state-of-the-art object detection model, for precise cow posture estimation. Similarly, (Kosourikhina, Kavanagh, Richardson, & Kaplan, 2022) validated DeepLabCut's

accuracy using high-precision devices, and (Gong et al., 2022) developed a YOLOv4 model specifically for cow detection and behavior tracking, achieving impressive results across varying conditions. This growing body of work demonstrates the effectiveness of combining deep learning and machine learning to automate livestock behavior monitoring, leading to improvements in both animal welfare and farm productivity.

The authors (Perez and Toler-Franklin, 2023) conducted an in-depth study of convolutional neural networks (CNNs) for action recognition and pose estimation, focusing on categorizing animal behaviors as observed from videos. In their study, they showed the importance of CNNs in this context and demonstrated all how these networks could classify and decode multiple animal activities defined in different videos. In recent years, numerous studies have focused on applying machine learning and pose estimation techniques to understand animal behavior more effectively. These studies emphasize automating behavior monitoring to reduce manual labor, improve accuracy, and enhance animal welfare.

The aim of this study is to leverage advanced pose estimation techniques and deep learning models to develop an automated system for real-time cattle behavior monitoring. We extracted key posture data, such as sitting, standing, eating, aggressive, sit-with-leg-tie, and let-go-of-the-tail behaviors, to identify patterns. Our approach focuses on analyzing these behaviors in real time, enabling early detection of health issues to improve livestock management and animal welfare.

Materials and Methods

Data Acquisition and Manual Observation

Our research focused on analyzing cow behavior using video datasets collected from three dairy farms in Multan, Pakistan. Each farm housed approximately 10 cows, and the data collection period spanned one month. The cows were recorded across different lactation stages to capture a diverse range of behaviors. Each farm was equipped with Nikon DSLR D5500 camera a high-definition camera, capturing video continuously for 3 hours per day. A total of 21 videos were recorded, each lasting approximately 15 minutes. While a larger dataset was collected during this time, we selectively used 16 videos for this study. We extract and focus on key behavioral instances, such as eating, standing, sitting, sit with leg tie, let go of the tail, walking, and aggressive actions. So, it reduces the total video dataset length to approximately 30 minutes. This selection enabled us to focus on high-quality, relevant footage for the analysis of pose estimation and behavior classification.

Ethical Considerations

All procedures involving animals complied with institutional guidelines for animal welfare and ethical treatment. The study involved non-invasive video recordings of cows in their natural environment. Although the work did not involve direct physical intervention, we followed ethical protocols to minimize any potential stress during the video recording process.

Preprocessing Video Data

The video data was processed by extracting individual frames at a consistent frame rate using the Napari platform. Napari is an open-source, that let you to view and work with multi-dimensional image viewer in Python. It is useful for the task like annotating or analyzing the large and complex images. For example, video recording of animal behaviors. It allows researchers to browse the complex images to mark specific area of interest. Each frame was annotated manually to identify key body parts, such as the nose, eye, and body center, which

were essential for accurate pose estimation. To improve the model's generalization to different viewing angles and lighting conditions, we applied data augmentation techniques such as random rotations, flips, and scaling.

Frame Extraction and Annotation

Video data was converted into individual frames using the Napari platform. Each frame was carefully labeled to identify specific key points on the cow's body, which was crucial for high accuracy in pose estimation.

- Frame Extraction: Videos were split into a series of frames at a consistent frame rate using the Napari platform.
- Annotation: Each frame was meticulously annotated to label the key body parts of the cows, essential for training the pose estimation model.

Data Augmentation

Data augmentation techniques, such as random rotations, flips, and scaling, were applied to enhance the model's robustness. Random rotations expose the model to various orientations, enabling it to learn rotational invariance (Shorten & Khoshgoftaar, 2019). Flipping introduces symmetrical variations, which are critical for generalizing across mirrored contexts (Krizhevsky et al., 2012). Scaling adjusts object sizes to mimic changes in distance, aiding the model in learning scale-invariant features (Simonyan & Zisserman, 2014). Collectively, these augmentations increase training data diversity, reduce overfitting, and improve the model's generalization under varying conditions (Goodfellow et al., 2016).

Pose Estimation

Pose estimation is an important area in computer vision in which the task is to predict the locations of some key points on an object (cow in this case). In our study, we developed a deep learning-based model for pose estimation that can accurately identify key points on the cow's body, including the nose, eyes, ears, and body center. For this task, we employed a residual neural network (ResNet)-based architecture, which is widely used for human anatomy localization and adapted it for bovine pose estimation. The model was trained on a dataset of labeled cow images, where each image had predefined ground truth key points for body landmarks. Our approach aims to minimize the error between the predicted key points and the ground truth through supervised learning.

Pose estimation in cattle behavior analysis involves several critical steps that ensure accurate and effective monitoring of protective behaviors. The process begins with the extraction of key frames from videos capturing the protective behavior of cows. This step is essential as it identifies the most relevant frames that depict significant movements or actions, reducing the volume of data that needs to be processed and enhancing the focus on crucial behavior patterns.

1. There are 4 main steps of pose estimation model:
2. Extract key frames of cow protective behavior.
3. Manually mark cow's head, legs, and tail in different colors.
4. Train a neural network to predict body part locations.
5. Extract body part locations in cattle behavior videos.

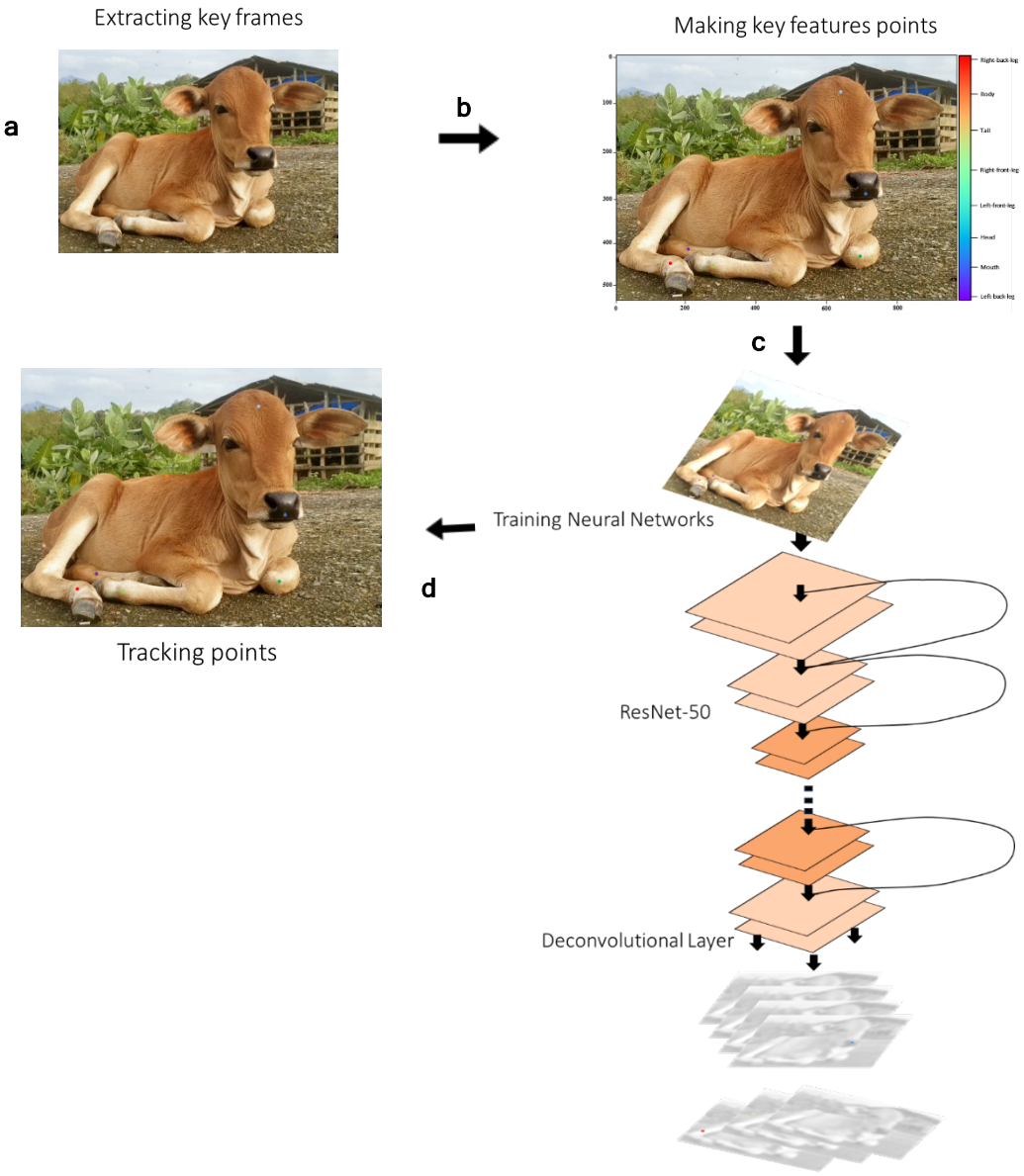

Figure 1: Pose Estimator. a. Extract key frames highlighting pose features indicative of cow protective behavior. b. Manually annotate body parts, employing distinct colors to identify the cow's head, legs, and tail. c. Develop and train a deep neural network capable of predicting body part locations from images, involving the determination of pixel probabilities. d. Apply the trained neural network to analyze cattle behavior videos, extracting accurate body part locations for further analysis.

Once the dataset is prepared with the marked key frames, a neural network is trained to predict the locations of the cow's body parts. This training process involves feeding the network with the annotated images and allowing it to learn the patterns and spatial relationships between different body parts. The neural network utilizes this training to develop a robust understanding of how the body parts move and interact during protective behaviors. The final step involves using the trained neural network to extract the body part locations from videos of cattle behavior. By applying the model to new video data, the system can automatically identify and track the

positions of the head, legs, and tail in real-time. This automated extraction enables researchers and farmers to monitor cattle behavior more efficiently and accurately, providing valuable insights into animal welfare and health.

Model Architecture Description

Input Layer:

The input to our pose estimation model consists of raw image frames extracted from video recordings of cow behaviors. These frames are pre-processed to ensure they have a consistent size and format suitable for further processing by the model. The images are typically resized to a fixed resolution (224x224 pixels) and normalized to have pixel values between 0 and 1.

Convolutional Layers:

The core of the feature extraction process is based on a deep convolutional neural network (CNN) architecture inspired by ResNet-50. ResNet-50 is a widely used model for image recognition tasks and is known for its ability to learn rich feature representations from images. The convolutional layers in our model are structured as follows:

- **Initial Convolution Layer:**

This layer applies a series of convolutional filters (kernels) to the input images. The filters are designed to capture various low-level features such as edges, textures, and color gradients.

Convolution operation: output = convolution (input, filter)

- **Residual Blocks:**

The hallmark of ResNet-50 is its use of residual blocks, which allow the network to learn residual functions with reference to the layer inputs. This is achieved through shortcut connections that bypass one or more layers.

Each residual block consists of a series of convolutional layers with ReLU activation functions and batch normalization.

Residual connection: output = input + F(input), where F represents the residual mapping.

The ResNet-50 architecture includes several stages, each with multiple residual blocks, progressively increasing the number of filters (e.g., from 64 to 128, 256, and 512).

Activation Function (ReLU):

Each convolutional layer is followed by a ReLU (Rectified Linear Unit) activation function, which introduces non-linearity into the model.

ReLU operation: ReLU(x) = max (0, x)

This activation function helps the model learn complex patterns by allowing it to capture non-linear relationships in the data.

Pooling Layers:

Pooling layers are interspersed between the convolutional layers to reduce the spatial dimensions of the feature maps, which helps in reducing computational complexity and controlling overfitting.

Max-Pooling:

Max-pooling layers select the maximum value from each pooling window (e.g., 2x2 window) and reduce the spatial dimensions by a factor of 2.

Max-pooling operation: output = max ($input_{2i:2i+2,\ 2j:2j+2}$)

This operation helps retain the most prominent features while discarding less important information.

Fully Connected Layers:

After the convolutional and pooling layers, the resulting feature maps are flattened into a one-dimensional vector, which serves as the input to the fully connected layers.

Dense Layers:

These layers consist of neurons that are fully connected to all the activations in the previous layer. They perform a weighted sum of the inputs followed by an activation function (typically ReLU).

Fully connected operation: output = ReLU (weights . Input +bias)

The final dense layer outputs a set of values corresponding to the predicted key point locations.

Output Layer:

The final output layer of the model is designed to produce heatmaps for each key point. Each heatmap indicates the likelihood of a key point being present at each location in the image.

Heatmap Generation:

The output layer applies a softmax activation function to produce a probability distribution over the spatial dimensions for each key point.

Softmax operation: softmax $(\mathbf{z_i}) = \frac{e^{z_i}}{\sum j e^{z_j}}$

This ensures that the predicted key points are spatially localized, allowing the model to accurately detect the positions of key anatomical landmarks on the cows.

Training and Loss Function:

The model is trained using a combination of supervised learning techniques and a suitable loss function (e.g., mean squared error) that measures the difference between the predicted heatmaps and the ground truth key points.

During training, the model adjusts its weights to minimize the loss function, thereby improving its accuracy in key point detection.

By carefully designing and training this pose estimation model, we achieved high accuracy in detecting key points on cows' bodies, enabling precise behavior classification in subsequent steps. The detailed architecture ensures robustness to variations in lighting, cow positions, and other environmental factors, making it a reliable tool for real-time monitoring of animal behavior.

To evaluate the performance of the pose estimation model, several key metrics were employed:

- **Training and Validation Curves**: Analyzed to track model performance during training and detect potential overfitting.
- **Ground Truth vs Predicted Points Comparison**: Visualized to inspect the accuracy of key point prediction in specific test cases.

Pose Estimation Accuracy Calculation

To evaluate the accuracy of the pose estimation model, we used the Percentage of Correct Keypoints (PCK) metric. PCK measures how accurately the model predicts key anatomical points compared to the true positions. The PCK score is calculated as follows:

$\text{PCK}=\frac{1}{n}\sum_{i=1}^{n} \frac{|T_i - P_i|}{|T_i|} \times 100$

Where:

n is the number of key points,

T_i is the true position of key point i,

P_i is the predicted position of key point i.

This metric gives a percentage score that reflects the accuracy of the pose estimation for each key point across the dataset. A higher PCK score indicates better alignment between the predicted and true key point positions.

Trajectory Graph Generation and Behavior Classification

Pose trajectories for each body part were derived, and thresholds were applied to kinematic features such as speed, angle, and movement duration. Specifically, thresholds were set as follows: for speed, a minimum of 0.2 m/s was required to classify a behavior as "walking," while movements below 0.05 m/s were classified as "standing" or "sitting" based on pose configurations. Eating behavior was identified using both pose configurations and movement patterns associated with repetitive hand-to-mouth gestures, typically involving minimal displacement and angular changes below 15 degrees. Additionally, an angular threshold of 30 degrees was used to distinguish "aggressive" behaviors, as these often involve sharper, more dynamic movements compared to routine activities. These thresholds were fine-tuned using annotated data, ensuring accurate behavior classification by the model.

For the successful execution of this research, several software tools and platforms were employed, each serving a specific purpose in the data processing and analysis pipeline.

Libraries

- Python 3.8: The programming language that is used for model implementation
- TensorFlow: It is used to train and build the pose estimation model.
- Keras: It is used for building high-level neural networks, it's a user-friendly API, which work on the top of TensorFlow.
- OpenCV: It is used to process videos and frames extraction.
- Pandas: It is used for utilizing data manipulating and analysis.
- Scikit-learn: It is used to implementing machine learning models.
- Matplotlib and Seaborn: It is used for creating data visualization.
- Napari: It is an open-source, multi-dimensional image viewer used for labeling and annotating image frames extracted from videos.
- Jupyter Notebook: It is used for experimenting with data processing and machine learning model training.
-

Additional Tools

- VS Code: It is used for primary code editor.
- Git: Used for version control and collaboration.

Results

Data Collection and Preprocessing

The dataset used in this study was meticulously curated to encompass a wide array of behaviors exhibited by cows. Each frame was annotated with key points to facilitate accurate pose estimation. Table 1 shows the dataset that diverse behaviors: eating, sitting with legs tied, standing, walking, and aggressive actions. The dataset consists of a total of 16 video recordings, which collectively span approximately 30 minutes. These recordings were converted into frames with a cumulative total of 52,780 frames.

Table 1. Dataset of 5 Behaviors

BEHAVIOR	TOTAL VIDEO LENGTH	NUMBER OF VIDEOS	TOTAL FRAME COUNT
Eating	6 min 3 sec	3	10900
Sit-with-leg-tie	6 min 30 sec	2	11100
Standing	5 min 6 sec	4	9180
Walking	7 min 25 sec	4	13350
Aggression	4 min 35 sec	3	08250

Annotation using Napari

Figure 2 illustrates the annotated frame the video using Napari. Each yellow point represents a key point or an interested area of a cow, such as the tail base, back middle, body middle left, front right thigh, and various other points. These labeling helps to clearly identify what each dot represents. These annotations are accompanied by labels for clarity.

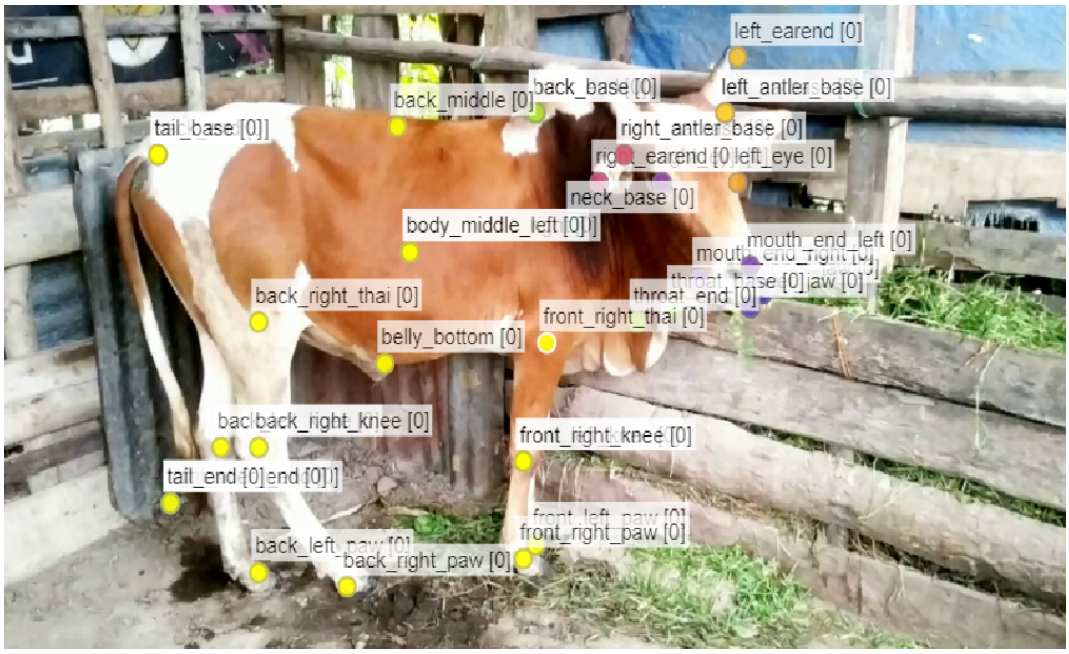

Figure 2: Annotated frame

Pose Estimation Accuracy

The marker-less pose estimation model demonstrated high accuracy in detecting key body points across different behaviors, achieving a PCK (Percentage of Correct Keypoints) score of 95%. This score indicates reliable pose estimation performance across the dataset.

Training and Validation Curves

The Graph in figure 3 shows how the training loss (blue line) and validation loss (red line) change over 20 epochs. The training loss keeps going down steadily, meaning the model is getting better at predicting the training data.

The validation loss also decreases, but with a few ups and downs, which is normal because it's tested on new data. The important thing is that both lines are moving downward, showing that the model is learning and performing well without overfitting.

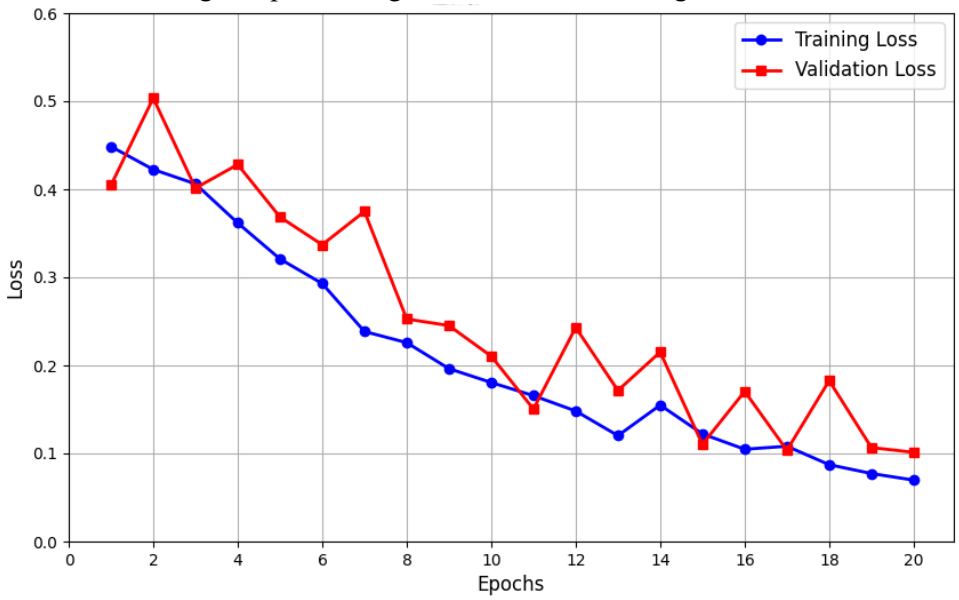

Figure 3: Training & Validation loss of Pose Estimator. The blue line represents the training loss, while the red line represents the validation loss.

Ground Truth vs Predicted Points Comparison

The model demonstrates high accuracy in predicting keypoints on unseen data as shown in figure 4.

The comparison between ground truth and predicted keypoints reveals minimal deviations across various body parts:

- **Cow Nose**: Predicted keypoints closely match the ground truth, indicating precise tracking.
- **Eyes**: Predictions for both eyes show minor variations in x and y coordinates but remain largely aligned with the ground truth.
- **Ears**: There is a slight discrepancy between the predicted and ground truth key points for the ears, suggesting potential improvement areas.
- **Body Center**: The central body key point is accurately detected, highlighting the model's robustness in identifying core features.

This confirms the model’s ability to generalize key features in real-world scenarios.

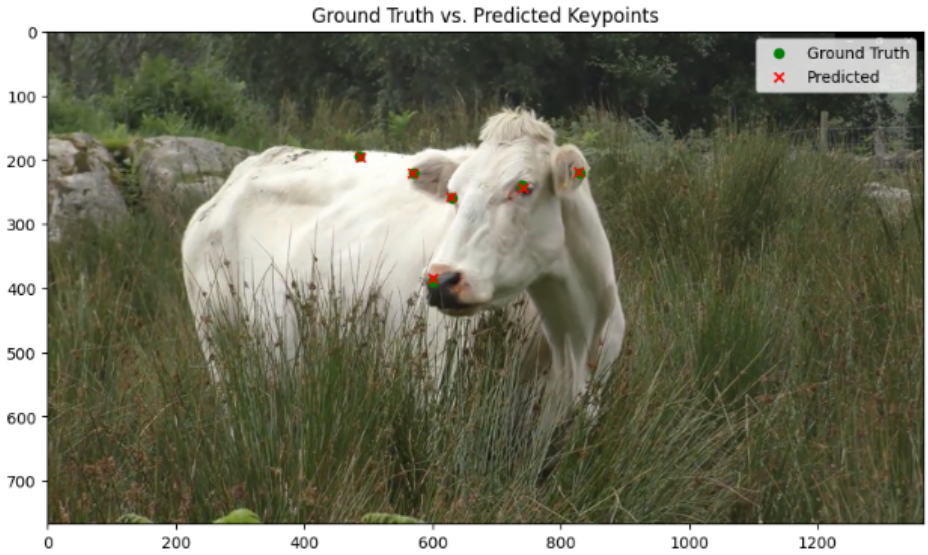

Figure 4: Visualization of Pose Estimator

X and Y Position Over Time

Figure 5 (a, b) shows track horizontal (X-axis) and vertical (Y-axis) movement of body parts over time, measured in pixels. Dashed lines represent horizontal movement, while solid lines depict vertical movement. Each line corresponds to a specific body part's movement over time, offering detailed insights into movement patterns.

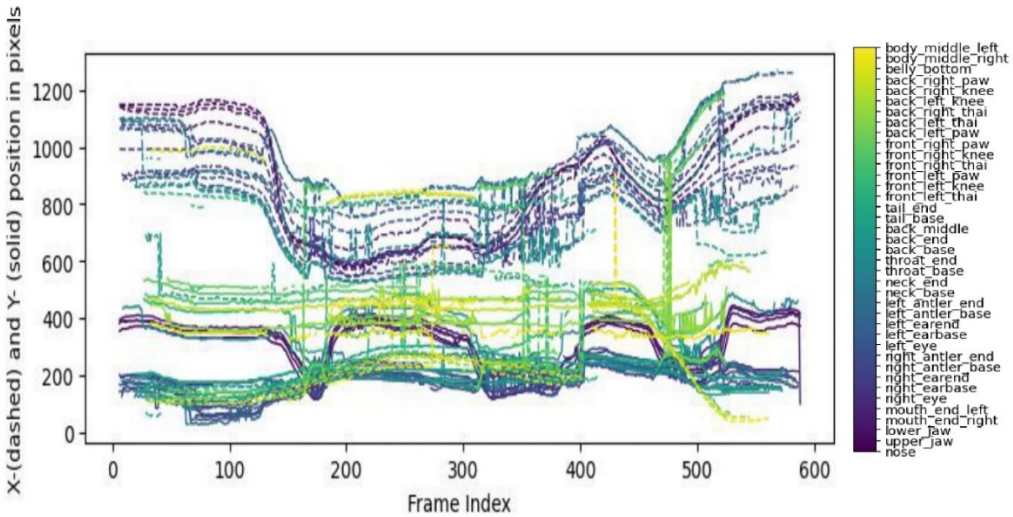

Figure 5 (a): Plot every body part across time (frame). Movement of a standing cow.

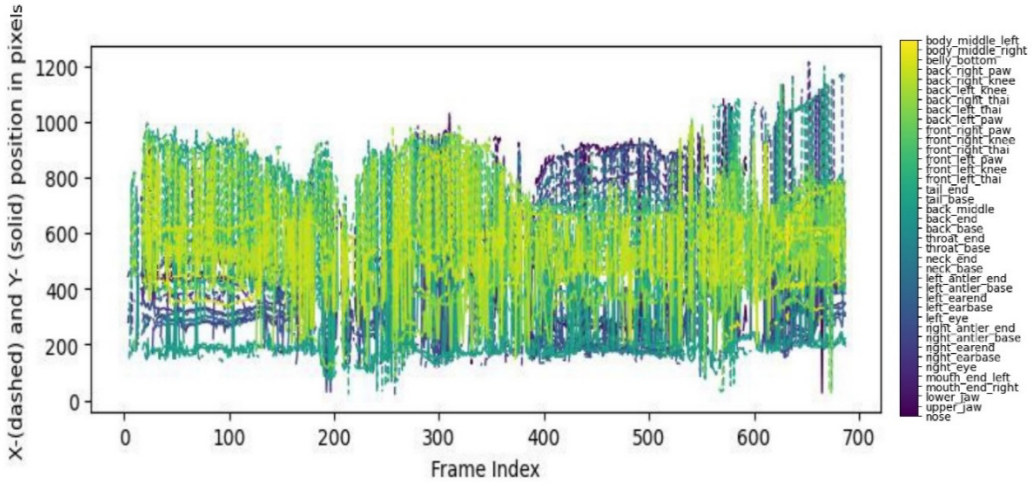

Figure 5 (b): Plot every body part across time (frame). Movement during aggressive behavior.

Likelihood Over Time

Figure 6 (a, b) shows the likelihood of accurately identifying body parts, with confidence levels ranging from 0 to 1 on the y-axis. Higher values indicate more reliable tracking, while fluctuations suggest uncertainty. The color-coded legend on the right corresponds to specific body parts, such as the nose, jaw, and limbs. These plots highlight the robustness and variability in the tracking performance across different body parts, which is crucial for accurate behavior classification.

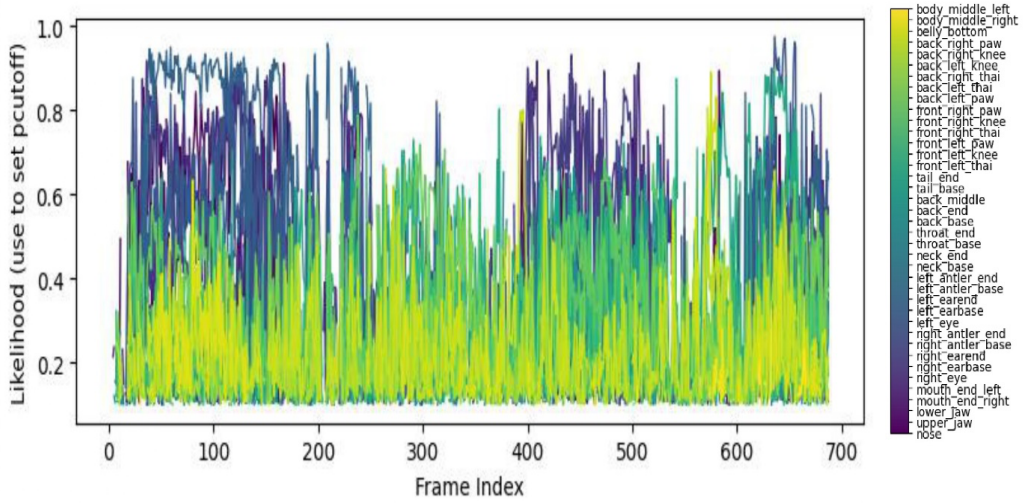

Figure 6 (a): Plot likelihood. The likelihood values of different key body parts over 700 frames of video.

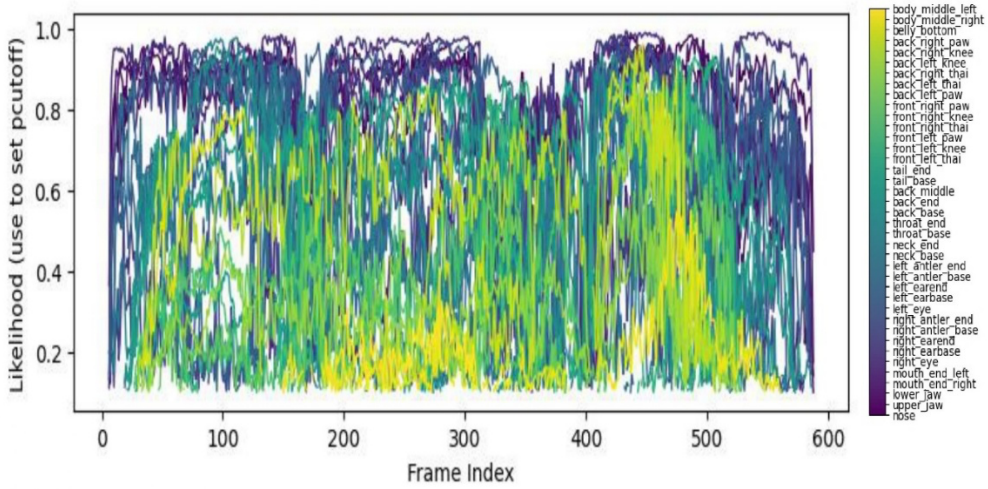

Figure 6 (b): Plot likelihood. The likelihood values of different key body parts over 600 frames of video.

Trajectory of Body Parts

This scatter plot visualizes the 2D trajectories of body parts in pixels. Different colors represent different body parts, and clusters of points highlight areas where specific parts linger. In Figure 7 (a, b) The spread of points reflects the overall range of movement.

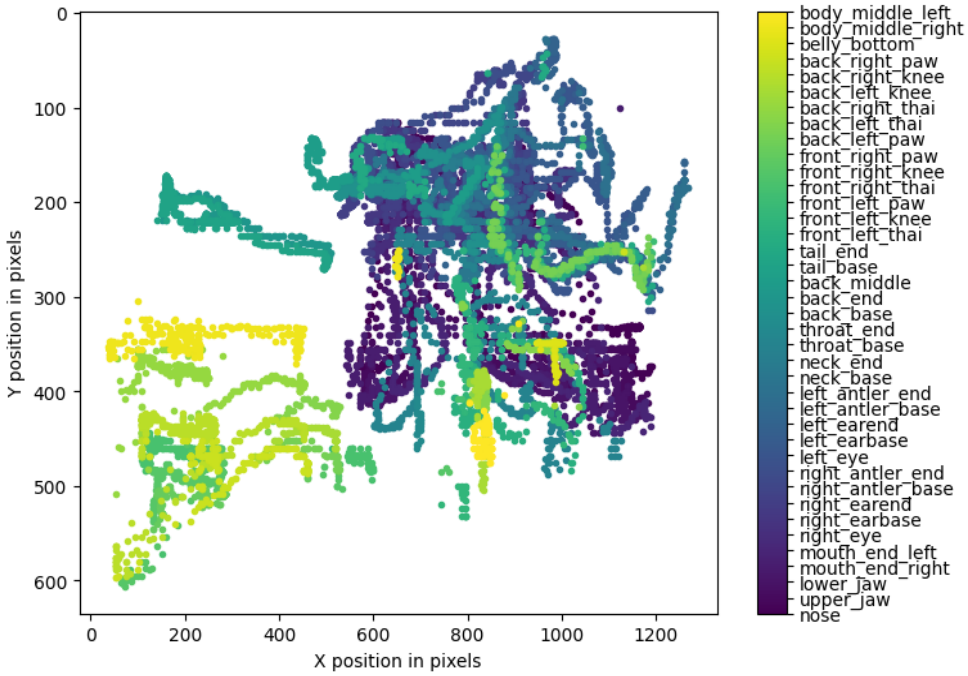

Figure 7 (a): Plot Trajectory

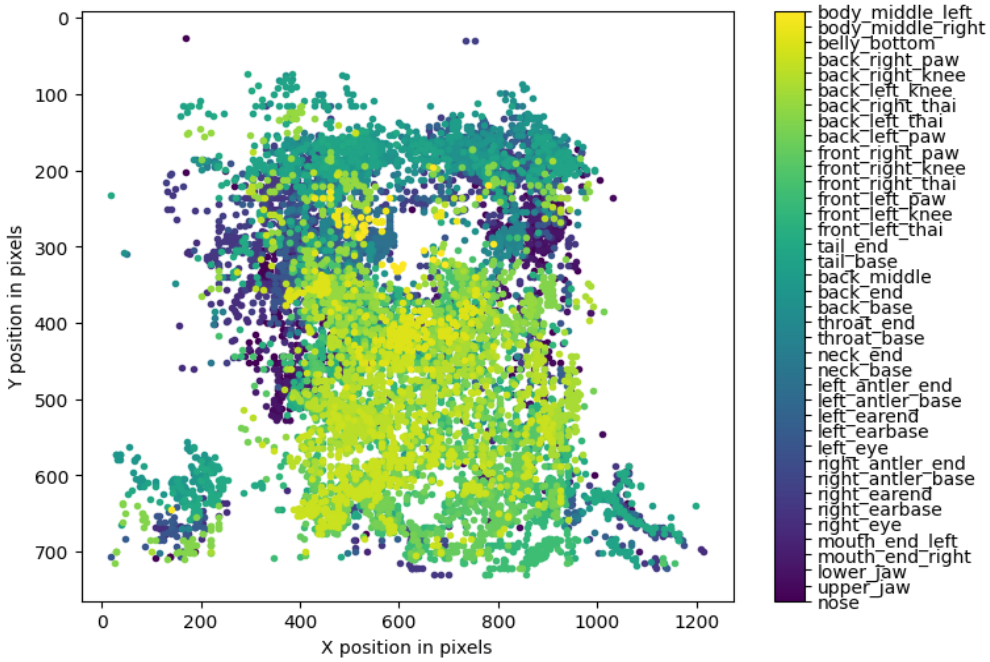

Figure 7 (b): Plot Trajectory

Behavior Classification

The behavior classification model was evaluated on the annotated dataset using precision, recall, and F1-score metrics. The classification was performed using both a decision tree and a random forest classifier. The decision tree classifier achieved 60% accuracy, struggling particularly with behaviors like eating due to a low number of distinguishing features.

In contrast, the random forest classifier significantly improved performance, achieving 96% accuracy across all behaviors, demonstrating the model's ability to handle complex datasets effectively. This robustness is underscored by the model's success in correctly classifying behaviors that the decision tree found challenging.

Table 2 summarizes the precision, recall, and F1-score results for each behavior. The results indicate that the model performed well across all behaviors, with precision values ranging from 0.90 to 0.94, recall values from 0.88 to 0.94, and F1-scores from 0.89 to 0.93. Specifically, the model achieved the highest F1-score for "Eating" and "Aggression" behaviors, both at 0.93, indicating a high level of accuracy in predicting these behaviors.

Table 2. Performance Metrics of Behavior Classifier Results

BEHAVIOR	PRECISION	RECALL	F1-SCORE
Eating	0.92	0.94	0.93
Sitting	0.90	0.88	0.89
Standing	0.93	0.91	0.92
Walking	0.91	0.89	0.90
Aggression	0.94	0.92	0.93

All results are expressed as percentages, with values representing the proportion of true positive classifications against the total predictions made by the model. For instance, the

"Eating" behavior achieved a precision of 0.92, a recall of 0.94, and an F1-score of 0.93, illustrating the model's effectiveness in accurately identifying instances of this behavior.

Behavior Classification Using Decision Tree

We initially used a decision tree classifier to classify cow behaviors into five classes in Figure 8: eating, sitting, standing, walking, and aggression. The dataset was divided into training, validation, and test sets, where 60% of the frames were allocated for training, and 20% each for validation and testing. The distribution was as follows:

- Eating: Training (6,540 frames), Validation (2,180 frames), Testing (2,180 frames)
- Sitting with Leg Tie: Training (6,660 frames), Validation (2,220 frames), Testing (2,220 frames)
- Standing: Training (5,508 frames), Validation (1,836 frames), Testing (1,836 frames)
- Walking: Training (8,010 frames), Validation (2,670 frames), Testing (2,670 frames)
- Aggression: Training (4,950 frames), Validation (1,650 frames), Testing (1,650 frames)

The cross-validation score provides a more realistic measure of the model's performance on unseen data. The confusion matrix (Figure 8) illustrates the performance of the model by displaying how often each behavior is correctly or incorrectly classified. The diagonal values represent the number of correct classifications for each behavior, with higher diagonal values indicating better performance for those classes. The confusion matrix also helps to understand the types of errors the model makes and identifies which classes it struggles to differentiate.

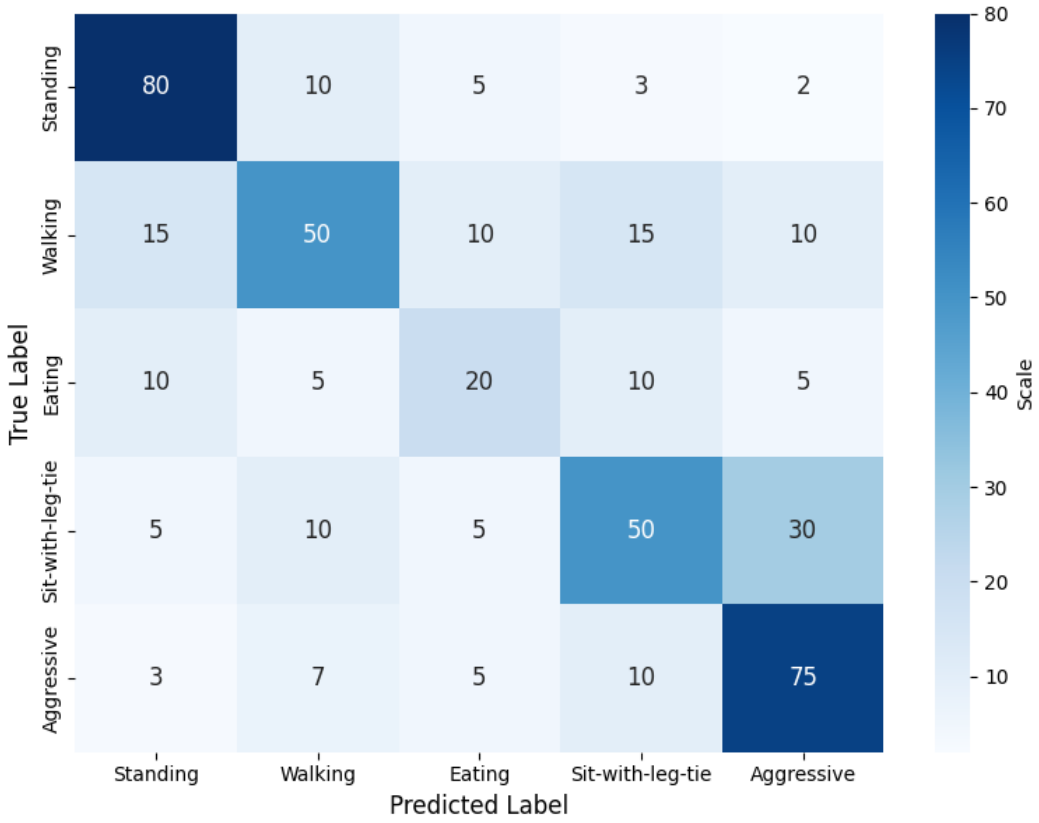

Figure 8: Confusion Matrix of Decision Tree. The model demonstrates high accuracy for 'Standing' and 'Aggressive' behaviors and low in eating

- **Standing:** The model achieves high accuracy for classifying 'Standing,' as shown by the strong diagonal value in Figure 8. However, there are some misclassifications, particularly with 'Walking,' suggesting that these behaviors may share similar features that lead to confusion.
- **Walking:** The accuracy for 'Walking' is moderate, with noticeable misclassifications into 'Standing' and 'Sit-with-leg-tie'. This implies that the model has difficulty distinguishing between these behaviors due to overlapping characteristics in their feature space.
- **Eating:** Eating shows significant misclassifications, likely due to less distinctive features or a smaller number of training samples. Figure 8 illustrates how frequently 'Eating' is confused with other behaviors, which impacts overall accuracy.
- **Sit-with-leg-tie:** While the model performs relatively well for 'Sit-with-leg-tie,' there are significant misclassifications with 'Walking' and 'Aggressive' suggesting some overlap in features among these actions.
- **Aggressive:** The model accurately classifies ' Aggressive ' behaviors, as evidenced by high correct classifications in Figure 8. Nevertheless, some misclassifications still occur, highlighting areas for potential model refinement.

Diagonal Values:

- In standing the model correctly classifies 80 instances of 'Standing'.
- In walking the 50 instances are correctly classified.
- In eating, only 20 instances are correctly classified, indicating a significant challenge in recognizing this behavior.
- In let-go-of-the-tail, 50 instances are correctly classified.
- In aggressive, 75 instances are correctly classified.

Off - Diagonal Values:

- Standing Misclassified as Other Behaviors:
  'Standing' is misclassified as 'Walking' (10 times), 'Eating' (5 times), 'Sit-with-leg-tie' (3 times), and 'Aggressive' (2 times).
- Walking Misclassified as Other Behaviors:
  Misclassifications include 'Standing' (15 times), 'Eating' (10 times), 'Sit-with-leg-tie' ' (15 times), and 'Aggressive' (10 times).
- Eating Misclassified as Other Behaviors:
  'Eating' is misclassified as 'Aggressive' (5 times), 'Sit-with-leg-tie' (5 times), 'Walking' (10 times), and 'Standing' (5 times).
- Sit-with-leg-tie Misclassified as Other Behaviors:
  It is misclassified with 'Aggressive' (10 times), 'Eating' (10 times), 'Walking' (15 times), and 'Standing' (3 times).
- Aggressive Misclassified as Other Behaviors:
  Misclassified include 'Sit-with-leg-tie' (30 times), 'Eating' (5 times), 'Walking' (10 times), and 'Standing' (2 times).

Training and Cross-Validation Performance

Decision tree classifier achieved an overall accuracy of 60%, as depicted in Figure 9. The training accuracy is 1.0 for all training sizes, indicating that the model is perfectly fitting the training data, which is a sign of overfitting. In a well-generalized model, training accuracy should be high but not perfect. This discrepancy suggests that the model does not generalize

well to unseen data. The low cross-validation accuracy Figure 9 highlights overfitting, as the model captured patterns in the training data but struggled to apply these patterns to new data.

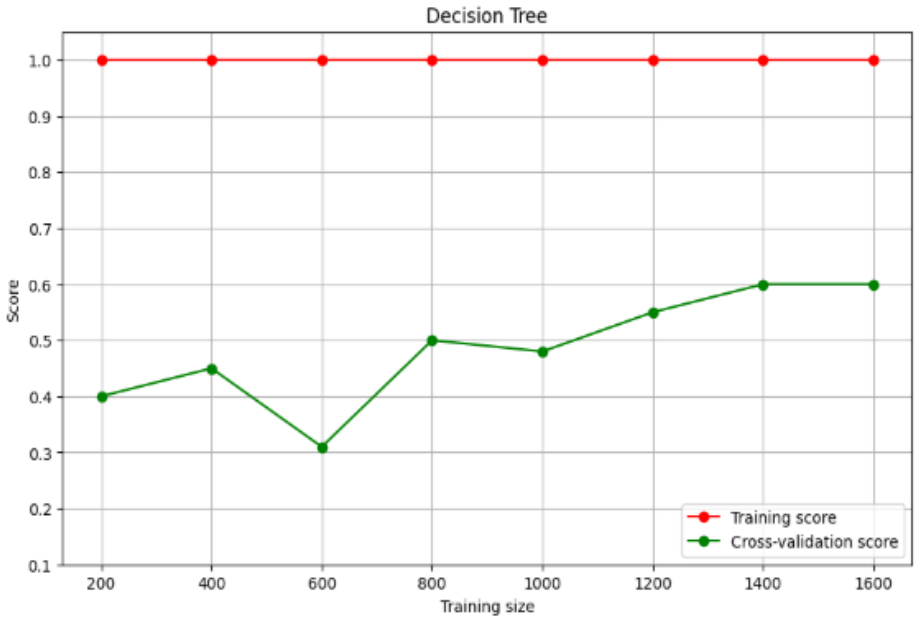

Figure 9: Training and Cross-Validation Accuracy of the Decision Tree Model. The red line represents the training score, the green line shows the cross-validation score.

Behavior Classification Using Random Forest

A Random Forest model was implemented to classify five animal behaviors: Standing, Walking, Eating, Aggressive, and Sitting. The model was trained using 100 decision trees, and the data set contained a total of 2,000 samples, each represented by 10 features. The classifier achieved an overall accuracy of 83.25%, based on both the training and cross-validation datasets.

The confusion matrix, Figure 10 provides a detailed breakdown of the classifier's performance across the five behaviors. Each row represents the true class of a behavior, and each column shows the predicted class. Diagonal elements reflect correct predictions, while off-diagonal elements highlight misclassifications

- **Standing:** This behavior was identified with an accuracy of 92% (73 out of 79 true instances), with minimal confusion with Aggressive behavior.
- **Walking:** The model in walking behavior achieves impressive classification accuracy of 98%, correctly identifying 81 out of 82 instances.
- **Eating:** Eating behavior had a moderate classification accuracy of 74%, with some confusion with Aggressive and Standing behaviors, which caused a drop in performance for this class.
- **Aggressive:** It was identified with an accuracy of 79%, although some instances were confused with Eating and Sitting.
- **Sit-with-leg-tie:** This behavior identified with 92% accuracy, correctly predicting 81 out of 88 true cases, with some misclassification as Walking.

Overall, Walking and Standing behaviors were classified with the highest accuracy, while Eating and Aggressive behaviors exhibited more overlap, reflecting the model's challenge in distinguishing between these dynamic behaviors.

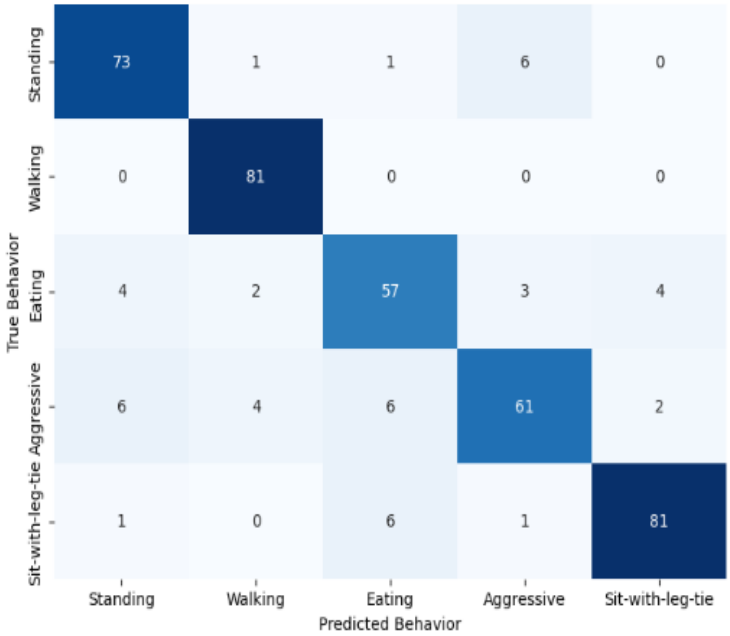

Figure 10: Confusion Matrix of Random Forest for Predicting Cow Activities. Y-axis: true behaviors; x-axis: predicted behaviors.

Diagonal Values:

- In standing the model correctly classifies 73 instances of 'Standing'.
- In walking the 81 instances are correctly classified.
- In eating, only 57 instances are correctly classified, indicating a significant challenge in recognizing this behavior.
- In let-go-of-the-tail, 61 instances are correctly classified.
- In aggressive, 81 instances are correctly classified.

Off - Diagonal Values:

- Standing Misclassified as Other Behaviors:
  'Standing' is misclassified as 'Eating' (4 times), 'Aggressive' (6 times), and 'Sit-with-leg-tie' (1 times).
- Walking Misclassified as Other Behaviors:
  Misclassifications include 'Standing' (1 times), 'Eating' (2 times), and 'Aggressive' (4 times).
- Eating Misclassified as Other Behaviors:
  'Eating' is misclassified as 'Standing' (1 times), 'Aggressive' (6 times), and 'Sit-with-leg-tie' (6 times).
- Aggressive Misclassified as Other Behaviors:
  Misclassified include 'Standing' (6 times), 'Eating' (3 times), and 'Sit-with-leg-tie' (1 times).
- Sit-with-leg-tie Misclassified as Other Behaviors:
  It is misclassified with 'Eating' (4 times), and 'Aggressive' (2 times).

Training and Cross-Validation Performance

The model's learning performance is depicted in Figure 11, which shows the learning curve for both the training and cross-validation scores. The red curve represents the model's accuracy on the training dataset, and the green curve represents its performance on unseen data during cross-validation.

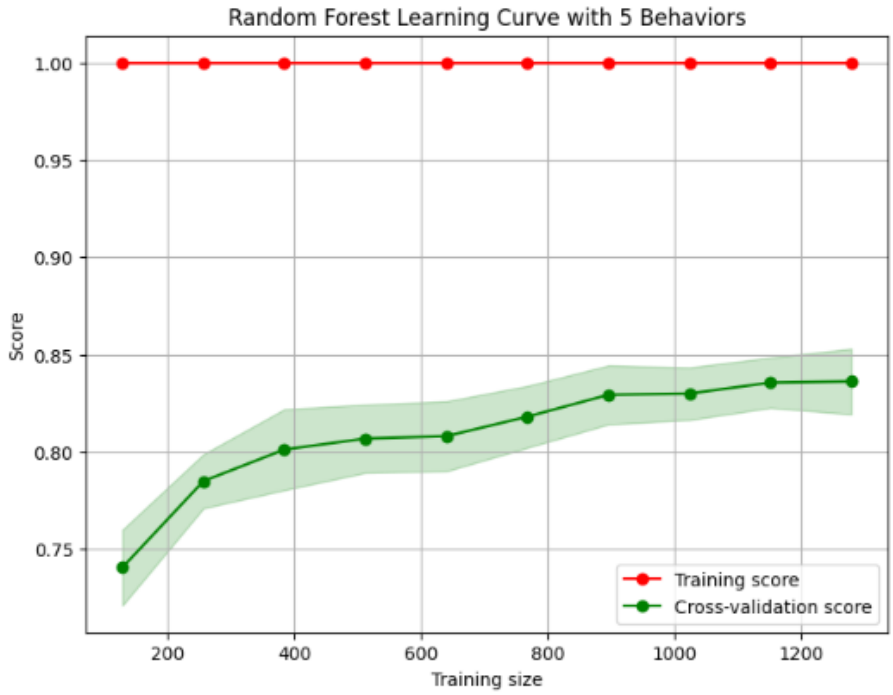

Figure 11: Random Forest Learning Curve for 5 Behavior Classification. The red line represents the training. The green line shows the cross-validation accuracy.

The training accuracy reached near 100%, demonstrating the model's ability to memorize the training data effectively.

The cross-validation score started at approximately 75% and steadily increased as more data was used for training. It reached a maximum of 83.62%, indicating strong generalization performance but also a slight gap between training and cross-validation accuracy, suggesting some degree of overfitting.

Expanding the Behavior Set with Random Forest

To further validate the robustness of the random forest algorithm, we expanded our dataset by adding two more behaviors: 'Sitting' and 'let-go-of-the-tail'. The expanded dataset now consisted of seven classes, which posed an additional challenge for the model to differentiate between these various behaviors. The frame distribution for these two new behaviors was as follows:

- **Sitting**: Training (4,500 frames), Validation (1,500 frames), Testing (1,500 frames)
- **Let-go-of-the-tail**: Training (3,800 frames), Validation (1,267 frames), Testing (1,267 frames)

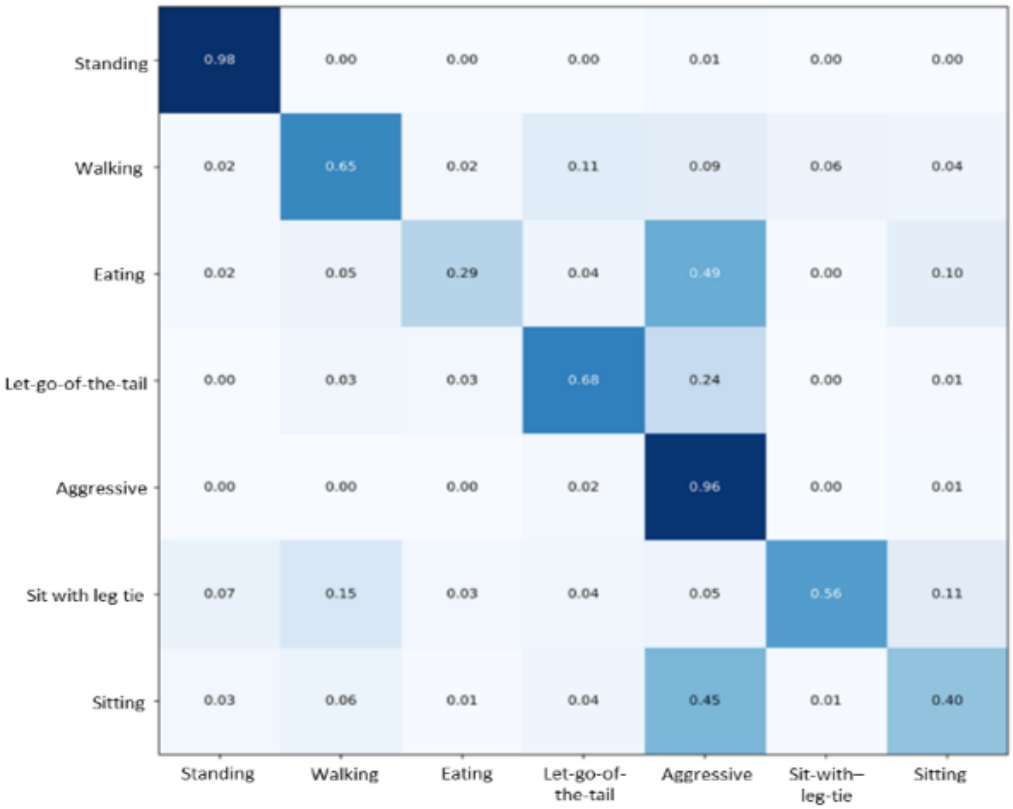

Figure 12: Cross-Validation Confusion Matrix of Random Forest for 7 Behavior Classification.

The figure 12 matrix shows a confusion matrix for a classification model that identifies seven different behaviors: Standing, Walking, Eating, Let-go-of-the-tail, Aggressive, sit with leg tie, and sitting. Each cell shows the proportion of times an actual behavior (rows) was classified as one of the possible behaviors (columns). The diagonal cells represent correct classifications, while off-diagonal cells represent misclassifications. This confusion matrix illustrates the performance of the Random Forest model across seven behaviors, with values normalized to represent classification accuracy. The diagonal elements show correct classifications, with the highest accuracies for "Standing" 98% and "Aggressive" 96%. Off-diagonal values represent misclassifications, such as "Eating" being confused.

Training and Cross-Validation Performance

The training and cross-validation graph in figure 13 demonstrates the model's learning performance. The training score remains consistently high, close to 1.00, showing that the model learns the training data very well. However, the cross-validation score starts lower but gradually improves as the training size increases, stabilizing around 0.96 with a larger dataset. This indicates that the model generalizes well but benefits from more data to reduce overfitting and improve its performance on unseen examples.

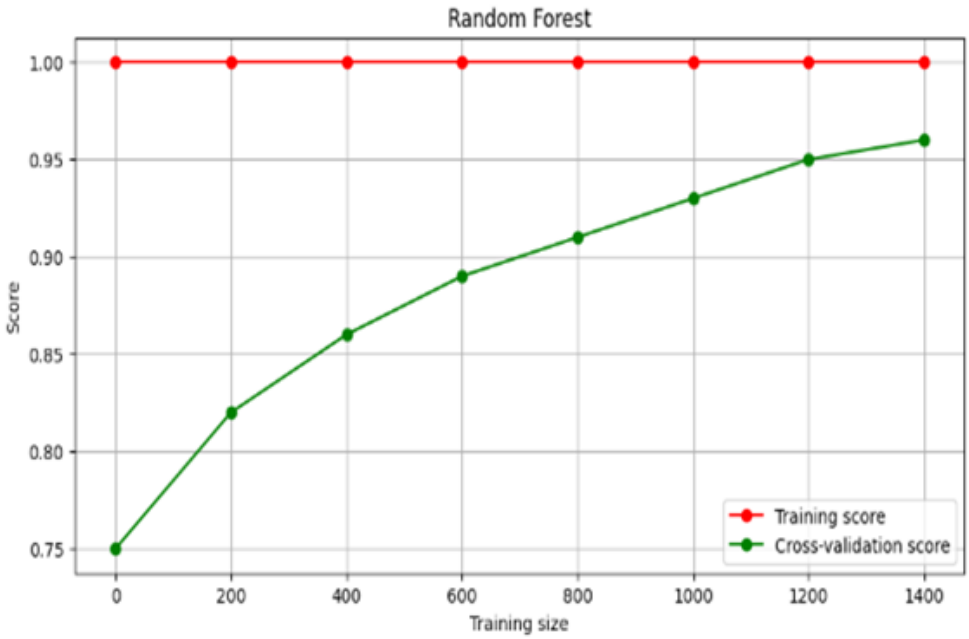

Figure 13: Random Forest Learning Curve for 7 Behavior Classification. training accuracy (red) and cross-validation accuracy (green) as a function of training size.

Classification Results on Real-Time

Figure 14 illustrates the real-time application of our pose estimation-based behavior classification system. The video captures a cow in its natural habitat, and our model processes the cow's skeletal pose to classify its behavior dynamically. In this case, the system identifies the behavior as "standing with eating" with high confidence, based on the extracted key body points and their movement patterns.

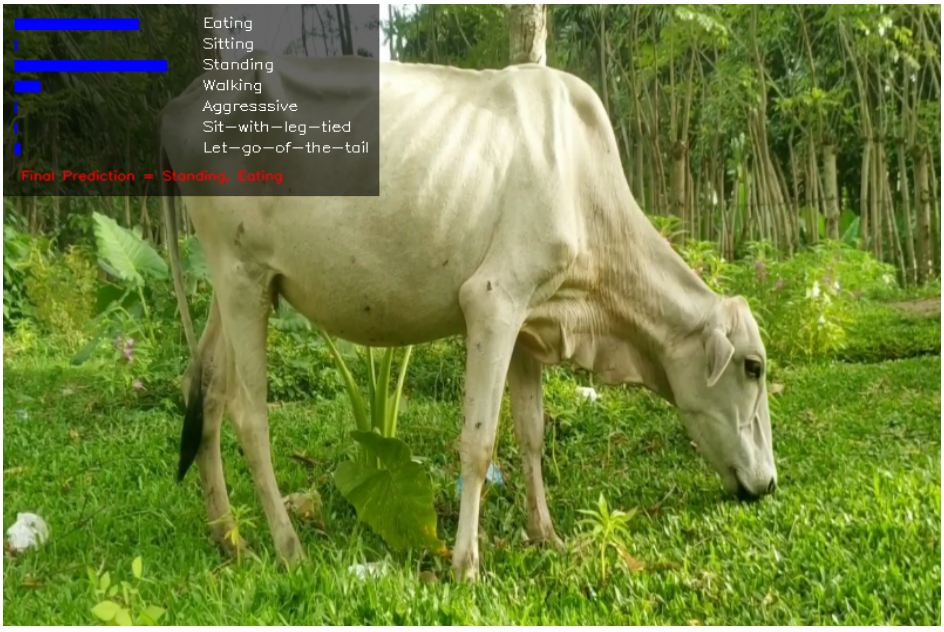

Figure 14: Video Classification Result. Behavioral prediction result for a cow, using a video-based classification model.

The live annotations, visible in the form of probability bars on the left side of the video frame, display the likelihood of various behaviors, with "standing" emerging as the most probable behavior. These predictions are updated frame by frame, showcasing the system's capacity to operate in real-world, uncontrolled environments. Figure 14 demonstrates the practical utility of our method in behavior recognition tasks, where real-time monitoring of livestock behavior can be crucial for early detection of health issues and effective management of animal welfare. By relying on pose estimation, the system ensures accurate and efficient classification without the need for extensive manual labeling, further emphasizing its applicability in large-scale cattle monitoring.

Discussion

The primary objective of this research was to develop and validate an automated system for monitoring cow behavior using the ResNet-50 pose estimation model. Our results underscore the effectiveness of modern deep learning techniques in classifying various cow behaviors, which is crucial for improving animal welfare and livestock management practices. The system was able to accurately identify and classify behaviors into seven categories, enhancing the efficiency and reliability of behavior monitoring compared to traditional, manual observation methods.

Traditional methods of analyzing animal behavior, as highlighted by (Kashiha et al., 2013) and (Guzhva, 2018) were labor-intensive and often prone to inconsistencies. In their study (Kashiha et al., 2013) used cameras and supervised learning to study pig behavior, with a focus on detecting water use and aggression, but required extensive manual data categorization. Similarly (Guzhva, 2018) developed a mathematical model to describe cow behavior, primarily focusing on physical traits rather than dynamic behaviors. While these studies laid the foundation for automated behavior monitoring, they were limited by manual data processing or a narrow scope of behavior types.

Our research advances these foundations by leveraging deep learning and computer vision. Using ResNet-50, we accurately identified key anatomical landmarks, allowing for the creation of comprehensive trajectory lines that capture complex movement patterns. While the pose estimator provides highly accurate landmark identification and trajectory generation, there is still room for improvement in the classification of certain behaviors, such as eating. The transition from a decision tree to a random forest classifier improved overall classification accuracy from 60% to 96%, underscoring the value of ensemble methods in managing the complexity and diversity of animal behavior data.

While the system performed well overall, certain behaviors, such as "eating" and "let go of the tail," showed lower classification accuracy. This aligns with findings from (Wang et al., 2018) who reported that combining multiple data sources (e.g., kinematics and location data) could improve behavior classification accuracy. Our system, primarily based on visual data, benefitted from examining several keypoints to enhance the robustness of behavior detection.The researchers (Mathis et al., 2018) introduced the DeepLabCut framework for pose estimation with minimal training data, proving that deep learning models can achieve high accuracy with less annotated data. Our use of ResNet-50 expands this innovation to larger animals like cows, whose complex movements require more detailed tracking.

A significant improvement in our system was the switch from decision trees to random forests. As noted by (Li et al., 2018) random forest classifiers outperform traditional models like decision trees and K-Nearest Neighbors, particularly in terms of generalization and accuracy for complex datasets. The random forest model reduced overfitting and improved classification performance, demonstrating the strength of ensemble methods for behavior recognition tasks.

However, the accuracy of our system is highly dependent on the quality of training data. This finding aligns with studies by (Nath et al., 2019) and (Labuguen et al., 2019), which emphasized

the importance of high-quality, annotated datasets in achieving optimal performance for pose estimation tasks. Expanding our dataset to include a wider range of behaviors and environmental conditions could further improve the robustness of our system.

The implications of this research are significant for the future of livestock management and animal welfare. Automated systems like ours could enable real-time monitoring of specific behaviors, such as sitting, sit-with-leg-tie, let-go-of-the-tail, standing, eating, and aggressive actions, which are key indicators of animal health and welfare. By reducing reliance on manual observation, our system offers a more objective and consistent method of tracking animal behavior, which could lead to improved productivity and welfare outcomes.

Limitations and Future Work

While our system demonstrated high accuracy in pose estimation, but in classification of behaviors there are areas for improvement. Specifically, behavior like "eating", were misclassified more frequently, likely due to the similarity between certain behaviors. Future work could explore integrating additional data sources, such as accelerometers or environmental sensors, to provide a more comprehensive understanding of cow behavior. Furthermore, although the random forest model performed well, it is computationally intensive, which may pose a limitation for real-time applications. Future studies could investigate more efficient algorithms or hardware acceleration techniques to address this challenge. Additionally, expanding the dataset to capture more diverse behaviors and conditions, as well as testing the system in uncontrolled, real-world environments, would enhance its generalizability.

References

Avanzato, R., Beritelli, F., & Puglisi, V. F. (2022, November). Dairy cow behavior recognition using computer vision techniques and CNN networks. In *2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)* (pp. 122-128). IEEE.

Chen, C., Zhu, W., & Norton, T. (2021). Behaviour recognition of pigs and cattle: Journey from computer vision to deep learning. *Computers and Electronics in Agriculture*, *187*, 106255.

Fujimori, S., Ishikawa, T., & Watanabe, H. (2020, October). Animal behavior classification using DeepLabCut. In *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)* (pp. 254-257). IEEE.

Gong, C., Zhang, Y., Wei, Y., Du, X., Su, L., & Weng, Z. (2022). Multicow pose estimation based on keypoint extraction. *PloS one*, *17*(6), e0269259.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. Chapter 7: Regularization for Deep Learning.

Guzhva, O. (2018). *Computer vision algorithms as a modern tool for behavioural analysis in dairy cattle* (No. 2018: 33).

Kashiha, M., Bahr, C., Ott, S., Moons, C. P., Niewold, T. A., Ödberg, F. O., & Berckmans, D. (2013). Automatic identification of marked pigs in a pen using image pattern recognition. *Computers and electronics in agriculture*, *93*, 111-120.

Kosourikhina, V., Kavanagh, D., Richardson, M. J., & Kaplan, D. M. (2022). Validation of deep learning-based markerless 3D pose estimation. *Plos one*, *17*(10), e0276258.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (pp. 1097-1105).

Labuguen, R., Bardeloza, D. K., Negrete, S. B., Matsumoto, J., Inoue, K., & Shibata, T. (2019, May). Primate markerless pose estimation and movement analysis using DeepLabCut. In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd international conference on imaging, vision & pattern recognition (icIVPR)* (pp. 297-300). IEEE.

Li, J., Kang, F., Zhang, Y., Liu, Y., & Yu, X. (2023). Research on Tracking and Identification of Typical Protective Behavior of Cows Based on DeepLabCut. *Applied Sciences*, *13*(2), 1141.

Li, J., Wu, P., Kang, F., Zhang, L., & Xuan, C. (2018). Study on the Detection of Dairy Cows' Self-Protective Behaviors Based on Vision Analysis. *Advances in Multimedia*, *2018*(1), 9106836.

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, *21*(9), 1281-1289.

Nath, T., Mathis, A., Chen, A. C., Patel, A., Bethge, M., & Mathis, M. W. (2019). Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nature protocols*, *14*(7), 2152-2176.

Pereira, T. D., Tabris, N., Matsliah, A., Turner, D. M., Li, J., Ravindranath, S., ... & Murthy, M. (2022). SLEAP: A deep learning system for multi-animal pose tracking. *Nature methods*, *19*(4), 486-495.

Perez, M., & Toler-Franklin, C. (2023). CNN-based action recognition and pose estimation for classifying animal behavior from videos: A survey. *arXiv preprint arXiv:2301.06187*.

Sakata, S. (2023). SaLSa: a combinatory approach of semi-automatic labeling and long short-term memory to classify behavioral syllables. *Eneuro*, *10*(12).

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6*(1), 1-48

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Tien, R. N., Tekriwal, A., Calame, D. J., Platt, J. P., Baker, S., Seeberger, L. C., ... & Kramer, D. R. (2022). Deep learning based markerless motion tracking as a clinical tool for movement disorders: Utility, feasibility and early experience. *Frontiers in Signal Processing*, *2*, 884384.

Wang, J., He, Z., Zheng, G., Gao, S., & Zhao, K. (2018). Development and validation of an ensemble classifier for real-time recognition of cow behavior patterns from accelerometer data and location data. *PloS one*, *13*(9), e0203546.

Wiltshire, C., Lewis-Cheetham, J., Komedová, V., Matsuzawa, T., Graham, K. E., & Hobaiter, C. (2023). DeepWild: Application of the pose estimation tool DeepLabCut for behaviour tracking in wild chimpanzees and bonobos. *Journal of Animal Ecology*, *92*(8), 1560-1574.

Wu, D., Wang, Y., Han, M., Song, L., Shang, Y., Zhang, X., & Song, H. (2021). Using a CNN-LSTM for basic behaviors detection of a single dairy cow in a complex environment. *Computers and Electronics in Agriculture*, *182*, 106016.