IYOLO-FAM: Improved YOLOv8 with Feature Attention Mechanism for Cow Behaviour Detection

Misbah Ahmad*†, Wenhao Zhang†, Melvyn Smith†, Ben Brilot*, and Matt Bell*
*Animal and Agriculture Department, Hartpury University, Gloucester, UK
Emails: {Misbah.Ahmad@hartpury.ac.uk, Ben.Brilot@hartpury.ac.uk, Matt.Bell@hartpury.ac.uk}
†Centre for Machine Vision, Bristol Robotics Laboratory, University of the West of England, Bristol, UK
Emails: {Misbah2.Ahmad@live.uwe.ac.uk, Wenhao.Zhang@uwe.ac.uk, Melvyn.Smith@uwe.ac.uk}

***Abstract*—We introduced IYOLO-FAM (Improved YOLOv8 with Feature Attention Mechanism) for detecting cow behaviours. By leveraging the robust YOLOv8 architecture improved with Feature Attention Mechanisms (FAM), Squeeze-and-Excitation (SE) blocks and data augmentation techniques, we enhanced the ability of the model to focus on salient features and generalize across a diverse farm environment. The experimental results demonstrated that IYOLO-FAM outperforms baseline YOLO models, achieving a mean Average Precision (mAP) of 88% at an IoU threshold of 0.5 and 70% across IoU thresholds from 0.5 to 0.95. These results highlighted substantial improvements over previous versions, particularly in detecting specific cow behaviours such as eating, lying, standing, and walking. The integration of SE blocks and FAM within the YOLOv8 framework proved effective in highlighting relevant features and enhancing detection accuracy, underscoring the significance of integrating advanced deep learning techniques with robust data augmentation techniques to tackle the challenges posed by a real-world farm environment. The proposed approach has the potential to benefit animal welfare in real-world applications, with future research focusing on integrating multimodal data. Additionally, real-world trials will validate the model's robustness and effectiveness in a practical farm environment.**

***Index Terms*—Precision Livestock Farming, Machine Learning, Deep Learning, Cow Behaviour Detection**

I. Introduction

Precision Livestock Farming (PLF) represents a transformative process in modern agriculture. It highlights the importance of accurately and timely detecting cow behaviours to provide effective herd management, optimise feeding traits, and enhance overall animal welfare. Accurate detection of cow behaviours is essential for several reasons, as it significantly enhances efficiency, sustainability, and productivity of livestock farming operations [1], [2]. By accurately detecting cow behaviours, PLS help minimise feed wastage and improve nutritional efficiency, leading to cost savings and better resource utilisation. Additionally, timely detection of health problems via behaviour changes allows prompt veterinary interventions, enhancing animal welfare.

Integrating technologies such as sensor networks and video surveillance further enhances the capabilities of PLF. It provides farmers and livestock managers with the tools to monitor and manage their herds effectively. Computer vision, particularly deep learning techniques, has revolutionised behaviour detection in livestock farming [3]. We presented an improvement to the YOLOv8 architecture by integrating Feature Attention Mechanisms, specifically Squeeze-and-Excitation (SE) blocks. SE blocks are used by first aggregating feature maps across spatial dimensions to create a channel descriptor (Squeeze) and then using this descriptor to produce a set of weights that rescale the original feature maps (Excitation). This process enables the developed model to emphasise on the most relevant features, improving its ability to detect variations in cow behaviours such as posture shapes, movement patterns, and texture. Our study aims to enhance the YOLOv8 model for more precise and effective cow behaviour detection using Cow dataset [4] & [5]. The main contributions of the work are as follows:

- To develop an improved deep learning model based on YOLOv8 for accurate detection of cow behaviours.
- To integrate SE blocks and FAM to improve model performance by dynamically adjusting feature map weights, thus emphasising key behaviour specific features.
- To train and fine tune the model on the Cow dataset, ensuring robustness and generalizability across the farm environment.
- To evaluate and compare the performance of the improved YOLOv8 against the baseline YOLO models using different evaluation metrics.

The work in this paper is arranged as follows: Section II provide a summary of existing methods for cow behaviour detection. Section III illustrates the Cow dataset used for experiments. Section IV details the methodology, including the improved YOLOv8 model, the addition of the SE blocks, FAM, and the training processes. Section V presents detection results and performance evaluation, comparing YOLOv8 with baseline YOLO models. Finally, Section VI concludes the paper and presents future work to enhance the proposed model's applicability.

II. Related Work

Research on livestock behaviour recognition has evolved significantly, transitioning from manual computer vision methods to cutting-edge deep learning techniques. Advancements in deep learning have brought a revolution in this field, showing enhanced capabilities in classification and detection methods [4] & [6].YOLACT algorithm used for monitoring the respiratory behaviour of cows, performing high accuracy in identifying resting states [7]. Deep learning algorithms have

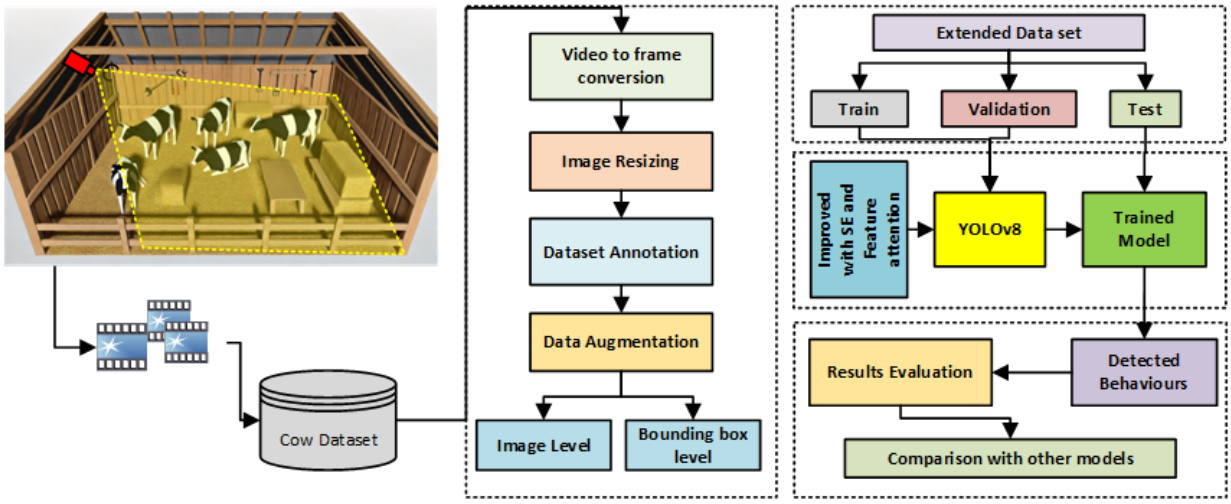

Fig. 1: The system setup for cow behaviour detection includes video-to-frame conversion, image resizing, dataset annotation, and data augmentation. The IYOLO-FAM model, enhanced with SE blocks and feature attention mechanisms, is trained on Cow dataset. Results are evaluated and compared with other models.

been increasingly utilized for livestock behaviour recognition, highlighting their applications in animal identification and behaviour detection [8]. For instance, a deep learning method for dairy cow rumination detection demonstrated the potential of video-based monitoring techniques to classify cow behaviours with high accuracy, paving the way for improved health monitoring and managing [9]. Moreover, introducing the cattle behaviour recognition approach using spatiotemporal information has allowed the recognition of multiple activities in video streams [10]. Another application of deep learning involves automatically detecting dairy cow feeding behaviour using facial images, improved by edge computing to present real-time monitoring abilities [11]. Authors in [12] developed a technique for facial expression recognition in pigs to evaluate on-farm welfare, highlighting the ethical significance of animal welfare and its impact on productivity. Further, methods focusing on cow tail detection and tracking have enhanced the accuracy of behaviour monitoring in precision livestock farming [13]. Furthermore, [4] developed a livestock activity monitoring system using a fine-tuned deep learning model for real-time multiclass cattle behaviour.

In summary, the transition from manual computer vision to deep learning for livestock behaviour detection has shown remarkable improvements in accuracy, real-time processing, and the capability to handle complex farm environments. These improvements collectively illustrate the potential of deep learning to improve livestock farming practices, providing better health, productivity, and welfare for the animals.

III. Data set

The dataset used in this work is collected at the Cow [5] & [4]. It consists of videos, which are then converted into images. Different cattle behaviours were captured within a barn environment, highlighting the diversity and complexity of real-time cattle activities, including lying down, eating, standing, and walking. These variations are important for comprehensive behaviour analysis. Each image was annotated with bounding boxes specifying the location and size of the object (cow) of interest. In total, 10000 images were used for the experiment. The dataset is used for training, validation, and testing is shown in Table I.

Activity	Training	Validation	Testing	Augmented	Total
Standing	2029	515	306	150	3000
Lying	1816	515	269	150	2750
Walking	1422	493	285	100	2300
Eating	1633	477	290	50	1950
Total	**6900**	**2000**	**1150**	**450**	**10000**

TABLE I: Dataset distribution with data augmentation.

IV. Methodology

The system setup, as outlined in Figure 1, starts with the basic preprocessing steps, which include video-to-image conversion, resizing the images, and manually labelling each frame for relevant behaviour categories. To avoid overfitting and achieve a more generalisable model, data augmentation is performed at two stages: image-level augmentation, which includes techniques such as rotation, scaling, and flipping, and bounding box-level augmentation, which involves modifying the sizes and positions of the bounding boxes around detected cows. After preprocessing, the dataset is split into three subsets: training, validation, and testing. The improved model is then used for training. This model is improved with SE blocks and FAM to enhance its performance. The following subsection explains the methodology step by step.

A. YOLOv8 for cow behaviour Detection

We used the state-of-the-art YOLO [14] model for cow behaviour detection. This model treats object detection as a

single regression problem. It directly maps image pixels to the coordinates of the bounding box and class probabilities. As illustrated in Figure 2, the input image is split into a grid of $S \times S$ by the network, and bounding boxes are predicted using each grid cell B. The corresponding confidence scores indicate the accuracy of the bounding box and the possibility of a target object. Following bounding boxes prediction and class probabilities, redundant boxes with lower confidence scores are removed using non-maximal suppression (NMS), which minimises the number of times the same object is detected. Consistent with the literature, we selected YOLOv8

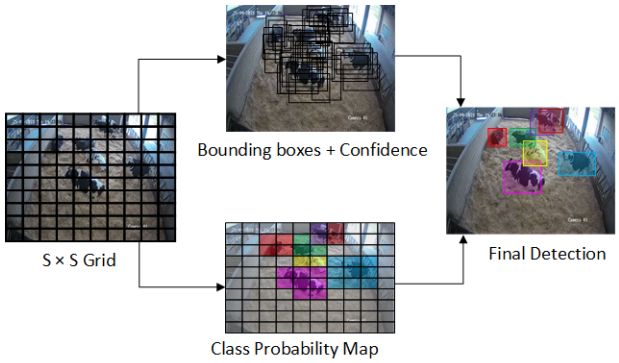

Fig. 2: YOLO Grid Division (S x S) for Cow Behaviour Detection.

as a baseline due to its robustness compared to other deep learning models [11], [15]. While various YOLO variants were tested, our primary focus was YOLOv8's performance in cow behaviour detection. YOLOv8, the latest iteration by Ultralytics [15], introduces new features for enhanced implementation, efficiency, and flexibility. YOLOv8 is highly adaptable, supporting various AI vision tasks such as detection, tracking, segmentaion and pose estimation. The architecture includes a backbone network for extracting initial features, a neck network for combining multi-scale features via Feature Pyramid Networks (FPNs), and a prediction output head for object detection. This design ensures high detection accuracy while maintaining computational efficiency.

Backbone Network: The backbone uses convolutional processes to extract features from RGB images at various scales, forming the foundation of YOLOv8's architecture.

Neck Network: Extracted features are combined in the neck network using FPNs, improving the robustness of the model in detecting objects at various scales.

Prediction Output Head: The head layer is used for the prediction of the target class, using detectors of different sizes to accurately identify small and large objects.

B. Improved YOLOv8 Architecture

To further improve detection accuracy and robustness for cow behaviour detection, our word integrated a feature attention mechanism [16]. Specifically Squeeze-and-Excitation (SE) blocks [17], into the YOLOv8s architecture. This improved version of YOLOv8s is designed to better focus on salient features within images, which is important for precise detection and classification. The improved YOLOv8 architecture is depicted in Figure 3. The significance of Figure 3 lies in its detailed representation of the improved YOLOv8 model, showcasing the addition of SE blocks and FAM. This integration is particularly useful as it enhances the capability of the model to focus on important features, such as specific body movements and postures of cows, which are crucial for accurate behaviour classification.

C. Squeeze-and-Excitation (SE) Blocks

As seen in Figure 3, SE blocks are intended to enhance the representative ability of the network. It allow model to carry out dynamic channel-wise recalibration of features. This is accomplished by using two primary procedures:

- **Squeeze:** The squeeze operation provide a channel description by using Global Average Pooling (GAP) and reducing global spatial information. Mathematically, it is represented as:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j) \tag{1}$$

  In Equation 1, $x_c(i,j)$ represents the value at position (i,j) in channel c, while H and W are the height and width of the feature map, respectively. The result, z_c, is the channel descriptor for channel c, which captures the essential information of the channel by averaging over its spatial dimensions.
- **Excitation:** Within the SE block, the excitation step includes a fully connected layer with 16 neurons (FC(r=16)), followed by a non-linear transformation and a sigmoid activation. This process is crucial for recalibrating the feature maps. The equations used in this process are:

$$e = \delta(W_1 z) \tag{2}$$

$$s = \sigma(W_2 e) \tag{3}$$

  In Equation 2, e represents the first fully connected layer's output, where δ denotes the ReLU activation function. Here, W_1 is the weight matrix applied to the squeezed channel descriptor z. Equation 3 uses σ, the sigmoid activation function, to produce the final modulation weight s by transforming the intermediate output e using the weight matrix W_2. These weights are used to reweight the feature maps, focusing on the most informative features.

By integrating SE blocks into our model, we enhanced the model to focus on significant features within the cow images, such as specific behavioural postures or movements. This channel-wise recalibration ensures the model can better distinguish between different cow behaviours, improving detection accuracy.

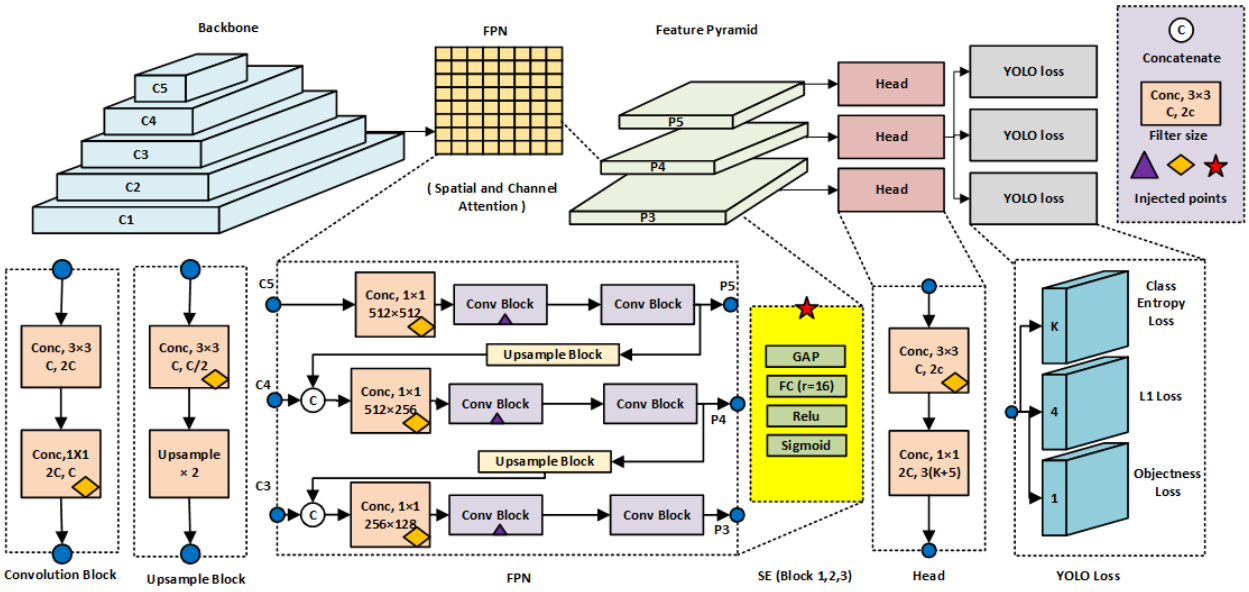

Fig. 3: The IYOLO-FAM architecture, integrating SE blocks and FAM at specific injected points, indicated by diamond, star, and triangle symbols, within the standard YOLOv8 structure. The diamond represents 3x3 convolution layers for spatial feature extraction, the triangle marks 1x1 convolution layers for dimensional adjustments, and the star indicates the integration of SE blocks for channel-wise feature recalibration. Spatial and channel attention mechanisms emphasize significant spatial locations. The architecture includes a backbone for initial feature extraction, an enhanced FPN for multi-scale feature aggregation, and a prediction head for object detection with composite loss functions for classification, localization, and attention.

D. Enhancements with Feature Attention Mechanism

Feature Attention Module:

- **Spatial Attention:** This approach involves creating an attention map A_s to highlight important spatial areas within feature maps. The mathematical formulation is given by:

$$A_s = \sigma(f_7^{(s)}(\text{concat}(\text{AvgPool}(X), \text{MaxPool}(X)))) \quad (4)$$

Equation 4 describes how A_s, the spatial attention map, is estimated. Here, $f_7^{(s)}$ represents a convolution operation with a kernel size of 7. The input X undergoes average pooling ($\text{AvgPool}(X)$) and max pooling ($\text{MaxPool}(X)$), which are concatenated and transformed by the convolution operation, followed by a sigmoid activation σ. The important spatial regions are highlighted within the feature maps by using this.

- **Channel Attention:** This method reweights the significance of various channels in the feature maps by assigning distinct weights to each channel. It is mathematically represented by:

$$A_c = \sigma(W_0^{(c)}(\text{AvgPool}(X)) + W_1^{(c)}(\text{MaxPool}(X))) \quad (5)$$

In Equation 5, A_c is the channel attention map. The average and max pooled features of input X are transformed by the weight matrices $W_0^{(c)}$ and $W_1^{(c)}$, respectively. The sum of these transformations is given through a sigmoid function σ to create the channel attention map, which enables the model to prioritize the most informative features across channels.

The Feature Attention Mechanism ensures that the model focuses on important spatial regions and channels within the cow images, enhancing the detection and classification of cow behaviours by emphasizing the most important features.

E. Backbone and Neck Modifications

- **Additional Convolutional Layers:** These layers were added to the backbone to extract more detailed features, enhancing the representational capacity of the model. This allows the network to capture finer details that are crucial for distinguishing between different cow behaviours.
- **Enhanced Feature Pyramid Networks (FPN):** The FPN in the neck was improved to ensure better multi-scale feature extraction. This enhancement allows the network to effectively combine low-level and high-level feature information, which is important for detecting objects of various sizes and scales.

Table II details the differences between the standard YOLOv8 and the improved YOLOv8 architecture. For instance, integrating SE blocks after certain layers improves the model's capability to focus on important features within the cow images. The strides used in different layers (e.g., Stride 2 for downsampling and Stride 1 for maintaining spatial resolution) ensure that the model captures both high-level and fine-grained details, which is essential for accurate behaviour detection.

TABLE II: YOLOv8 Vs Improved YOLOv8

Component	YOLOv8	Improved YOLOv8
Backbone		
Stem Layer	3 × 3, 32, Stride 2	3 × 3, 32, Stride 2
P1	3 × 3, 64, Stride 2	3 × 3, 64, Stride 2
Stage 1	3 × 3, 64, Stride 1	3 × 3, 64, Stride 1
P2	3 × 3, 128, Stride 2	3 × 3, 128, Stride 2
Stage 2	3 × 3, 128, Stride 1	3 × 3, 128, Stride 1
P3	3 × 3, 256, Stride 2	3 × 3, 256, Stride 2
SE Block 1	None	Integrated after P3: GAP, FC (r=16), ReLU, FC, Sigmoid
Stage 3	3 × 3, 256, Stride 1	3 × 3, 256, Stride 1
P4	3 × 3, 512, Stride 2	3 × 3, 512, Stride 2
SE Block 2	None	Integrated after P4: GAP, FC (r=16), ReLU, FC, Sigmoid
Stage 4	3 × 3, 512, Stride 1	3 × 3, 512, Stride 1
P5	5 × 5, 512, Stride 1	5 × 5, 512, Stride 1
SE Block 3	None	Integrated after P5: GAP, FC (r=16), ReLU, FC, Sigmoid
Neck (FPN)		
FPN	Standard FPN	Enhanced FPN with Spatial and Channel Attention
TopDown Layer 1	3 × 3, 256, Stride 1 -	3 × 3, 256, Stride 1 (with Attention Mechanism)
Concat	-	-
UpSample	-	-
TopDown Layer 2	3 × 3, 256, Stride 1 -	3 × 3, 256, Stride 1 (with Attention Mechanism)
Concat	-	-
UpSample	-	-
TopDown Layer 3	3 × 3, 256, Stride 1 -	3 × 3, 256, Stride 1 (with Attention Mechanism)
Concat	-	-
DownSample	-	-
BottomUp Layer 1	3 × 3, 512, Stride 1	3 × 3, 512, Stride 1
Concat	-	-
DownSample	-	-
BottomUp Layer 2	3 × 3, 1024, Stride 1	3 × 3, 1024, Stride 1
Head		
P3	1 × 1, 256, Stride 1	1 × 1, 256, Stride 1
P4	1 × 1, 512, Stride 1	1 × 1, 512, Stride 1
P5	1 × 1, 1024, Stride 1	1 × 1, 1024, Stride 1
Bbox	3 × 3, 256, Stride 1	3 × 3, 256, Stride 1
Cls	3 × 3, 256, Stride 1	3 × 3, 256, Stride 1

- **P1, P2, P3, P4, P5:** Different stages of the backbone network where feature maps are extracted.
- **SE Block 1, 2, & 3:** GAP, FC (r=16), ReLU, FC, Sigmoid, integrated after Layer P3, P4 & P5.
- **TopDown Layers:** Layers that upsample higher-level feature maps and merge them with lower-level feature maps.
- **BottomUp Layers:** Layers that downsample and process feature maps for higher-level feature extraction.
- **Cls:** Classification layer that predicts class probabilities.
- **Bbox:** Bounding box regression layer that predicts bounding box coordinates.

F. Training Process

The training involves optimising a composite loss function and utilising specific optimisation techniques to enhance the performance of the model in detecting and classifying cow behaviours. Key components of the training process include:

- **Composite Loss Function:** The composite loss function combines classification loss, localisation loss, and attention loss to ensure accurate object classification, bounding box prediction, and focus on important features. It is expressed as:

$$L_{total} = L_{cls} + L_{loc} + L_{att} \tag{6}$$

Equation 6 defines the total loss L_{total}, which is the sum of classification loss L_{cls}, localization loss L_{loc}, and attention loss L_{att}. This combination ensures a balanced optimisation of the above model ability to classify objects, predict bounding boxes, and focus attention on significant features.

Classification Loss: This part of the composite loss is calculated using the binary cross entropy loss. It calculates the difference between the true class labels y_c and predicted probabilities p_c:

$$L_{cls} = -\sum_{c \in \text{classes}} [y_c \log(p_c) + (1 - y_c) \log(1 - p_c)] \tag{7}$$

In Equation 7, the loss estimates how well the model's predicted probabilities match the true labels across classes c.

Localisation Loss: This loss uses the smooth L1 loss function to calculate the error between predicted b_i and ground truth $\hat{b}_i$ bounding box coordinates:

$$L_{loc} = \sum_{i \in \text{boxes}} \text{Smooth}_{L1}(b_i - \hat{b}_i) \tag{8}$$

In Equation 8, Smooth_{L1} is the smooth L1 loss function. This loss penalises large errors in bounding box predictions while allowing for small errors to be less significant.

Attention Loss: This component ensures that the model effectively learns to focus on significant features by utilising the mean squared error (MSE) loss between predicted and target attention weights. It is given as:

$$L_{att} = \sum_{i \in \text{features}} \text{MSE}(A_i - \hat{A}_i) \tag{9}$$

In Equation 9, A_i are the attention weights predicted by the model, $\hat{A}_i$ are the target attention weights, and MSE is the MSE loss function. This loss encourages the model to learn attention patterns, highlighting the most important features in the cow images. The inclusion of the attention loss component ensures that the model effectively learns to focus on significant features within the cow images, improving the overall detection and classification accuracy.

- **Optimisation Techniques:** The Adam optimiser was used with an initial learning rate of 0.001. and batch

size of 32. It was trained over 150 epochs. Utilising the Adam Optimiser helps adjust the learning rate dynamically during training, which is crucial for achieving optimal performance without overfitting, especially given the complexity of cow behaviour detection tasks.

The integration of the feature attention mechanism, SE blocks, and modifications to the backbone and neck components significantly enhance the detection capabilities of YOLOv8s. A detailed comparison between YOLOv8 and improved YOLOv8 is provided in Table II. The attention mechanisms and SE blocks enable the model to focus on the most important regions of the image and the most informative features, leading to improved accuracy in detecting and classifying cow behaviours. Improved multi-scale feature extraction ensures robustness across different object sizes and scales, making the model more reliable in practical farm environments. These enhancements result in better performance metrics, such as higher mAP and more accurate Behaviour-specific detection rates, as demonstrated in this work.

V. Experimental Results

The experiments were conducted using a Core i7 3 GHz CPU and 32 GB of RAM on a Windows 11 PC. The training was performed using Google Colab, leveraging GPU capabilities to expedite the process. The implementation and training utilised PyTorch [18], with support from Ultralytics [15]. The dataset was split into a 70% training set, a 20% testing set, and a 10% validation set as discussed in Table I. The training set also included some background images (negative samples that do not contain the target object, a cow), which helped to reduce false positives.

A. Training Results

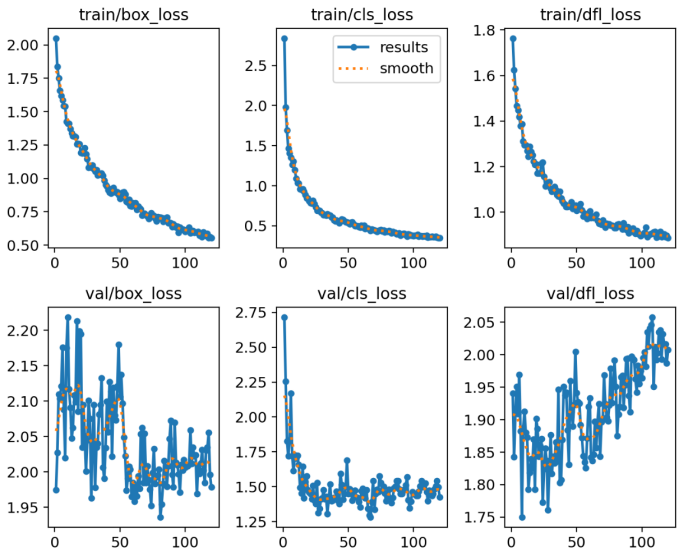

Fig. 4: Training and Validation Loss Curves for YOLOv8 Model Over 150 Epochs.

Training and validation loss of the model over 150 epochs is shown in Figure 4. Each subplot shows valuable information about the learning dynamics of the above model and its performance on the validation set. The x-axis in all subplots represents the number of epochs, ranging from 0 to 150, where each epoch describes one complete pass through the whole training set. The y-axis means the loss values, which show the model performance in terms of how well it is learning to make predictions. The subplots display the following loss metrics: The box regression loss starts at approximately 2.0 and steadily declines to around 0.5 by the 150th epoch for training (train/box_loss), indicating that the model is progressively improving its precision in predicting bounding box coordinates. For validation (val/box_loss), the loss begins at around 2.2, fluctuates slightly, and eventually decreases to about 1.95. This overall downward trend, despite some oscillations, suggests that the ability of models to generalize bounding box predictions to unseen data is improving. The classification loss shows a reduction during training (train/cls_loss), starting at about 2.5 and dropping to below 0.5 by the 150th epoch, indicating that the model is effectively learning to classify behaviours. The validation classification loss (val/cls_loss) decreases from approximately 2.75 to about 1.5, with some variations. The general downward trend suggests that the model's classification performance on the validation set is also improving over time, though not as smoothly as in the training data. The distribution focal loss consistently decreases during training (train/dfl_loss), from around 1.8 to about 1.0, showing that the model is becoming better at predicting focal distributions. In contrast, the validation focal loss (val/dfl_loss) starts at approximately 2.05 and shows a fluctuating upward trend, ending around 1.85. This could indicate challenges in generalizing the focal distribution predictions to the validation set, possibly hinting at slight overfitting towards the end of the training. Overall, the comparison between training and validation loss revealed that while the model is learning effectively, there are signs of slight overfitting, particularly in the focal loss towards the end of the training. This suggests that adjustments such as early stopping, learning rate decay, or additional regularization could be beneficial to enhance generalization.

B. Detection Results

The detection results of IYOLO-FAM are shown in Figure 5. The image compared the predicted and true labels for cow behaviour classes detected by the model. The image is arranged into four rows in a grid style, where each row pair shows a set of images true labels (top) and matching predicted labels (bottom). Among the activity classifications given to the cows in the first row pair are "Eating," "Lying," "Standing," and "Walking." The corresponding second row shows the model's predicted labels. Colour-coded labels indicating the activity and a confidence score (e.g., "Eating 0.8") are placed inside each bounding box. This confidence score, usually obtained from the output probabilities of the model's final layer, demonstrates how well the model makes its predictions.

Fig. 5: Comparative Analysis of True vs. Predicted Labels for Cow Activities. The image displays three pairs of rows, each pair showing the true activity labels (top) and the predicted labels (bottom) for a series of images. Activities like 'Eating,' 'Lying,' 'Standing,' and 'Walking' are labelled with bounding boxes and confidence scores.

The contrast illustrates how accurate the model is in determining the cows' activities in most cases. The second-row pair similarly shows another batch of images with true labels on top and predicted labels below. This set highlights areas where the model performs well and areas needing improvement, indicated by mismatched labels or lower confidence scores. The confidence scores are important as they reflect the confidence level of the model in its predictions. For example, a score of 0.8 for "Eating" means there is an 80% probability that the detected activity is "Eating." It is a probability value outputted by the model's final layer, generally based on a sigmoid function in the case of classification tasks. This score predicts the probability that the detected activity belongs to the predicted class. The third-row pair continues the comparison, providing more images for analysis. The performance of the above model is assessed by checking the overlap and match between true and predicted labels. Any differences can guide further refinements in the model. The final row pair shows the last set of images, concluding the comparative analysis. The consistency of the model's predictions across various scenarios and environments within the images is important for evaluating robustness. Observations from the image show that the model generally demonstrated good accuracy in detecting and labelling cow activities.

C. Performance Evaluation

This section provided performance evaluation results of the IYOLO-FAM used for cow behaviour detection. The results are also compared with various YOLO models (YOLOv1 to YOLOv8) on different behaviour classes. Precision (P), Recall (R), and F1-score (F1) are used for evaluation, which are considered important in determining the significance of the object detection models. We also used Mean Average Precision (mAP) at different IoU thresholds to estimate the performance of models across different classes (eating, walking, lying, and standing). The mAP is calculated as follows:

$$\text{mAP} = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{10}$$

In the above Equation, n describes the number of IoU thresholds, and AP_i is the Average Precision at each threshold.

- **mAP@0.5** which is calculated at a threshold of 0.5, estimated across all classes.
- **mAP@0.5:0.95** is computed at thresholds ranging from 0.5 to 0.95 with a step size of 0.05, estimated across all classes.

Table III presented the comprehensive comparison of Precision (P), Recall (R), and F1-score (F1) for YOLO models from YOLOv1 to YOLOv8 and also our IYOLO-FAM. The comparison focuses solely on YOLO variants because YOLO models have been widely recognized for their excellent performance in real object detection applications. YOLO's successive versions, from YOLOv1 to YOLOv8, demonstrate incremental progress in detection accuracy, making it a suitable baseline for considering enhancements. Comparing only YOLO variants allows us to highlight the specific advancements within this family of models and clearly attribute improvements to architectural and optimization changes. Each successive version of YOLO shows enhancements in evaluation metrics reflecting the continuous advancements in the architecture and optimization techniques. For the Eating class, YOLOv1 starts with a Precision of 0.70, Recall of 0.68, and F1-score of 0.69. By YOLOv8, these values have increased to 0.83, 0.81, and 0.82. The IYOLO-FAM further improves these metrics slightly, with a Precision of 0.84, Recall of 0.82, and F1-score of 0.83. This shows a consistent and steady improvement in detecting the Eating class across the versions.

The YOLO models demonstrate a steady improvement in detecting the "Lying" class, beginning with YOLOv1, which recorded a Precision, Recall, and F1-score of 0.76 and 0.77, respectively. Significant enhancements were observed in YOLOv8, where these metrics increased to 0.91, 0.89, and 0.90. The IYOLO-FAM advanced these figures to 0.92, 0.90, and 0.91, respectively, highlighting a substantial improvement in identifying the "Lying" behaviour, making it one of the most accurately detected categories. For the "Standing" class, YOLOv1 initially achieved scores of 0.80, 0.78, and 0.79. YOLOv8 improved these to 0.92, 0.90, and 0.91, which were maintained in the IYOLO-FAM, indicating the growing efficiency of YOLO models in recognizing "Standing"

TABLE III: Comparison of Precision (P), Recall (R), and F1-score (F1) for YOLO models from YOLOv1 to YOLOv8

YOLO Version	Class	P	R	F1
YOLOv1	Eating	0.70	0.68	0.69
YOLOv1	Lying	0.78	0.76	0.77
YOLOv1	Standing	0.80	0.78	0.79
YOLOv1	Walking	0.72	0.70	0.71
YOLOv1	All Classes	0.75	0.73	0.74
YOLOv2	Eating	0.72	0.70	0.71
YOLOv2	Lying	0.80	0.78	0.79
YOLOv2	Standing	0.82	0.80	0.81
YOLOv2	Walking	0.74	0.72	0.73
YOLOv2	All Classes	0.77	0.75	0.76
YOLOv3	Eating	0.74	0.72	0.73
YOLOv3	Lying	0.82	0.80	0.81
YOLOv3	Standing	0.84	0.82	0.83
YOLOv3	Walking	0.76	0.74	0.75
YOLOv3	All Classes	0.79	0.77	0.78
YOLOv4	Eating	0.76	0.74	0.75
YOLOv4	Lying	0.84	0.82	0.83
YOLOv4	Standing	0.86	0.84	0.85
YOLOv4	Walking	0.78	0.76	0.77
YOLOv4	All Classes	0.81	0.79	0.80
YOLOv5	Eating	0.78	0.76	0.77
YOLOv5	Lying	0.86	0.84	0.85
YOLOv5	Standing	0.88	0.86	0.87
YOLOv5	Walking	0.80	0.78	0.79
YOLOv5	All Classes	0.83	0.81	0.82
YOLOv6	Eating	0.80	0.78	0.79
YOLOv6	Lying	0.88	0.86	0.87
YOLOv6	Standing	0.90	0.88	0.89
YOLOv6	Walking	0.82	0.80	0.81
YOLOv6	All Classes	0.85	0.83	0.84
YOLOv7	Eating	0.82	0.80	0.81
YOLOv7	Lying	0.90	0.88	0.89
YOLOv7	Standing	0.91	0.89	0.90
YOLOv7	Walking	0.84	0.82	0.83
YOLOv7	All Classes	0.87	0.85	0.86
YOLOv8	Eating	0.83	0.81	0.82
YOLOv8	Lying	0.91	0.89	0.90
YOLOv8	Standing	0.92	0.90	0.91
YOLOv8	Walking	0.85	0.83	0.84
YOLOv8	All Classes	0.88	0.86	0.87
IYOLO-FAM	Eating	**0.84**	**0.82**	**0.83**
IYOLO-FAM	Lying	**0.92**	**0.90**	**0.91**
IYOLO-FAM	Standing	**0.92**	**0.90**	**0.91**
IYOLO-FAM	Walking	**0.86**	**0.84**	**0.85**
IYOLO-FAM	All Classes	**0.89**	**0.87**	**0.88**

behaviours. Regarding the ”Walking” class, the metrics in YOLOv1 began at 0.72, 0.70, and 0.71. By the time YOLOv8 was developed, these numbers had improved to 0.85, 0.83, and 0.84, with the IYOLO-FAM making slight additional gains to 0.86, 0.84, and 0.85. The overall trend indicates that YOLO models are becoming increasingly adept at detecting more complex behaviours like ”Walking.” At the same time, simpler actions such as ”Standing” and ”Lying” remain relatively easier to predict.

Overall, the YOLO models demonstrate consistent improvements in all evaluated metrics (Precision, Recall, and F1-score) as they evolve from YOLOv1 to YOLOv8. The IYOLO-FAM shows slight yet noteworthy improvements over the standard YOLOv8 model, indicating that further optimizations and enhancements can yield even better detection performance. Precision and Recall values for each class show a balanced improvement, leading to a higher F1 score, which signifies that the models have improved in determining true positives and reducing false negatives and positives. The steady improvement across the YOLO versions highlights the effectiveness of iterative enhancements and architectural innovations.

The heatmap graph in Figure 6 visually summarises the performance metrics across different YOLO models and cow behaviour classes. This detailed comparison and visualization (e.g., the heatmap and trend graphs) empirically demonstrate specific improvements and their impact on performance. These visual tools help to identify areas for future research and optimization, especially in complex behaviours such as ’Walking’ and ’Eating’. The heatmap effectively highlights the strengths and weaknesses of every model, offering a clear comparison of their detection capabilities. Each cell represents the F1-score for a particular behaviour class detected by a specific YOLO model, with colour intensity ranging from light to dark to indicate low and high F1-scores, respectively. This visual representation facilitates the quick identification of trends and patterns. The heatmap shows a clear trend of improvement in F1-scores from YOLOv1 to YOLOv8, with IYOLO-FAM demonstrating the highest scores across most behaviour classes. For example, the F1-score for the lying class improved from YOLOv1 to IYOLO-FAM, indicating a marked enhancement in the ability of the model to detect this behaviour accurately. Similar improvements are observed in other behaviour classes, such as Standing and Walking, underscoring the effectiveness of iterative advancements in the YOLO architecture. Moreover, the heatmap allows for easy performance comparison across different behaviour classes. It is evident that some classes, like ’Standing’ and ’Lying’, are detected with higher accuracy compared to more dynamic behaviours like ’Walking’ and ’Eating’. This insight is valuable for researchers and practitioners focusing on further optimization and fine-tuning models for complex behaviours. Overall, the heatmap is an effective tool for visualizing and comparing the performance of various YOLO models, providing clear evidence of the progressive improvements achieved through each version.

The two figures below illustrate the performance of various YOLO models in terms of False Positive Rate (FPR) and True Positive Rate (TPR) across different classes of activities. As shown in Figure 7, there is a clear trend of decreasing FPR with newer versions of YOLO, indicating that newer models are better at reducing false alarms. Each class (Eating, Lying,

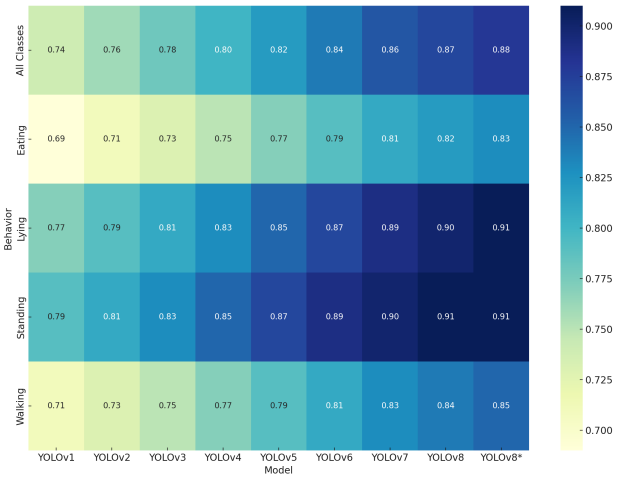

Fig. 6: A heatmap of the F1-score values for each YOLO version and Behaviour class.

Standing, Walking) shows a similar trend of decreasing FPR, with the highest FPR observed in the Eating class and the lowest in the Standing class. The overall FPR for all classes also demonstrates a reduction, indicating Improved model performance in minimizing false positives.

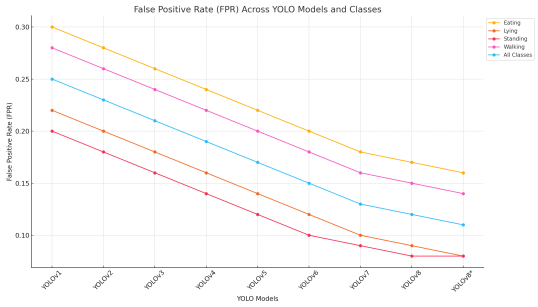

Fig. 7: The False Positive Rate (FPR) across different YOLO models and classes.

Conversely, Figure 8 demonstrates an increasing trend in TPR with newer YOLO versions, suggesting improved detection accuracy. Each class shows an improvement in TPR, with the Standing class achieving the highest TPR and the Eating class the lowest. The overall TPR for all classes improves consistently, reflecting the models' enhanced ability to identify the target activities correctly. These trends are consistent across all classes, highlighting the overall improvement in YOLO models over time. The class-specific trends reveal that 'Standing' generally performs best, while 'Eating' has the most room for improvement in both metrics. The combined cumulative performance for all classes mirrors these individual class trends, underscoring the advancements in YOLO model accuracy and reliability.

The comparison in Table IV is shown for mAP@0.5 and mAP@0.5:0.95 for YOLO models from YOLOv1 to YOLOv8 and our improved IYOLO-FAM or IYOLO-FAM. The mAP values comprehensively estimate the results of the above

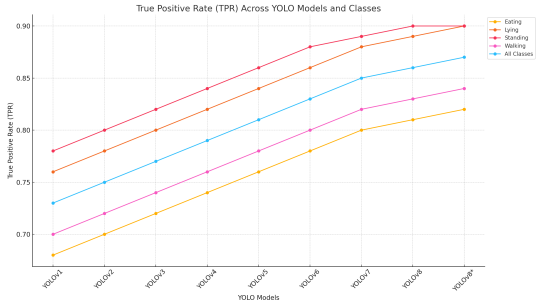

Fig. 8: True Positive Rate (TPR) across different YOLO models and classes.

model across different IoU thresholds. The results show a clear trend of improvement in mAP values as the YOLO models developed. YOLOv1 has a mAP@0.5 of 0.74 and a mAP@0.5:0.95 of 0.54. By YOLOv8, these values have increased to a mAP@0.5 of 0.87 and a mAP@0.5:0.95 of 0.68. The IYOLO-FAM model further improves these values to a mAP@0.5 of 0.88 and a mAP@0.5:0.95 of 0.70. These improvements indicate that the models are becoming more accurate and reliable in detecting objects across different IoU thresholds. In conclusion, the performance evaluation in Table IV illustrated the progressive enhancements in the YOLO models, showcasing their robustness and reliability in detecting various behaviours. The IYOLO-FAM improved performance underscores the potential for continued advancements in object detection technologies as well.

VI. Conclusion and Future Direction

This research introduced an Improved YOLOv8 model using the Feature Attention Mechanism (IYOLO-FAM) for cow behaviour detection. The model is improved using SE blocks and spatial channel attention modules. Experiments were conducted using a real Farm Cow behaviour dataset. The results demonstrated that the IYOLO-FAM shows good results compared to baseline models. The overall accuracy is 88% at (mAP@0.5) and 70% (mAP@0.5:0.95). The F1 scores for specific behaviours, such as eating, lying, standing, and walking, show substantial improvement, proving model effectiveness for different behaviour classes. Detection results improvements in behaviour detection showed that the improved model not only pushes the boundaries of cow behaviour detection but also paves the way for real-time monitoring applications in livestock management. The improved robustness and accuracy of the model make it a valuable tool for ensuring timely interventions and maintaining animal health and wellbeing. Future work could further refine the model to detect and track behaviours using video footage and explore integration with other deep-learning models.

Acknowledgment

We gratefully acknowledge the support and funding provided by John Oldacre Trust and Hartpury University under project No.HK009687 for this research endeavour. Their

TABLE IV: Comparison of mAP for YOLO (v1-v8) and improved IYOLO-FAM

YOLO Version	Class	mAP@0.5	mAP@0.5:0.95
YOLOv1	Eating	0.70	0.50
YOLOv1	Lying	0.77	0.55
YOLOv1	Standing	0.79	0.57
YOLOv1	Walking	0.71	0.53
YOLOv1	All Classes	0.74	0.54
YOLOv2	Eating	0.72	0.54
YOLOv2	Lying	0.79	0.57
YOLOv2	Standing	0.81	0.59
YOLOv2	Walking	0.73	0.55
YOLOv2	All Classes	0.76	0.56
YOLOv3	Eating	0.74	0.56
YOLOv3	Lying	0.81	0.59
YOLOv3	Standing	0.83	0.61
YOLOv3	Walking	0.75	0.57
YOLOv3	All Classes	0.78	0.58
YOLOv4	Eating	0.76	0.58
YOLOv4	Lying	0.83	0.61
YOLOv4	Standing	0.85	0.63
YOLOv4	Walking	0.77	0.59
YOLOv4	All Classes	0.80	0.60
YOLOv5	Eating	0.78	0.60
YOLOv5	Lying	0.85	0.63
YOLOv5	Standing	0.87	0.65
YOLOv5	Walking	0.79	0.61
YOLOv5	All Classes	0.82	0.62
YOLOv6	Eating	0.80	0.62
YOLOv6	Lying	0.87	0.65
YOLOv6	Standing	0.89	0.67
YOLOv6	Walking	0.81	0.63
YOLOv6	All Classes	0.84	0.64
YOLOv7	Eating	0.82	0.64
YOLOv7	Lying	0.89	0.67
YOLOv7	Standing	0.90	0.69
YOLOv7	Walking	0.83	0.65
YOLOv7	All Classes	0.86	0.66
YOLOv8	Eating	0.83	0.66
YOLOv8	Lying	0.90	0.69
YOLOv8	Standing	0.91	0.71
YOLOv8	Walking	0.84	0.67
YOLOv8	All Classes	0.87	0.68
IYOLO-FAM	Eating	**0.84**	**0.68**
IYOLO-FAM	Lying	**0.91**	**0.71**
IYOLO-FAM	Standing	**0.92**	**0.72**
IYOLO-FAM	Walking	**0.85**	**0.69**
IYOLO-FAM	All Classes	**0.88**	**0.70**

commitment to advancing scientific knowledge and innovative solutions in livestock management has been instrumental in the successful implementation of this work.

REFERENCES

[1] D. Berckmans *et al.*, "Automatic on-line monitoring of animals by precision livestock farming," *Livestock production and society*, vol. 287, pp. 27–30, 2006.

[2] R. Geers, B. Puers, V. Goedseels, P. Wouters *et al.*, *Electronic identification, monitoring and tracking of animals*. CAB international, 1997.

[3] T. A. Shaikh, T. Rasool, and F. R. Lone, "Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming," *Computers and Electronics in Agriculture*, vol. 198, p. 107119, 2022.

[4] M. Ahmad, W. Zhang, M. Smith, B. Brilot, and M. Bell, "Real-time livestock activity monitoring via fine-tuned faster r-cnn for multiclass cattle behaviour detection," in *2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2023, pp. 805–811.

[5] J. McDonagh, G. Tzimiropoulos, K. R. Slinger, Z. J. Huggett, P. M. Down, and M. J. Bell, "Detecting dairy cow behavior using vision technology," *Agriculture*, vol. 11, no. 7, p. 675, 2021.

[6] C. Chen, W. Zhu, and T. Norton, "Behaviour recognition of pigs and cattle: Journey from computer vision to deep learning," *Computers and Electronics in Agriculture*, vol. 187, p. 106255, 2021.

[7] D. Wu, M. Han, H. Song, L. Song, and Y. Duan, "Monitoring the respiratory behavior of multiple cows based on computer vision and deep learning," *Journal of Dairy Science*, vol. 106, no. 4, pp. 2963–2979, 2023.

[8] D. A. B. Oliveira, L. G. R. Pereira, T. Bresolin, R. E. P. Ferreira, and J. R. R. Dorea, "A review of deep learning algorithms for computer vision systems in livestock," *Livestock Science*, vol. 253, p. 104700, 2021.

[9] S. Ayadi, A. Ben Said, R. Jabbar, C. Aloulou, A. Chabbouh, and A. B. Achballah, "Dairy cow rumination detection: A deep learning approach," in *Distributed Computing for Emerging Smart Networks: Second International Workshop, DiCES-N 2020, Bizerte, Tunisia, December 18, 2020, Proceedings 2*. Springer, 2020, pp. 123–139.

[10] A. Fuentes, S. Yoon, J. Park, and D. S. Park, "Deep learning-based hierarchical cattle behavior recognition with spatio-temporal information," *Computers and Electronics in Agriculture*, vol. 177, p. 105627, 2020.

[11] Z. Yu, Y. Liu, S. Yu, R. Wang, Z. Song, Y. Yan, F. Li, Z. Wang, and F. Tian, "Automatic detection method of dairy cow feeding behaviour based on yolo improved model and edge computing," *Sensors*, vol. 22, no. 9, p. 3271, 2022.

[12] M. F. Hansen, E. M. Baxter, K. M. Rutherford, A. Futro, M. L. Smith, and L. N. Smith, "Towards facial expression recognition for on-farm welfare assessment in pigs," *Agriculture*, vol. 11, no. 9, p. 847, 2021.

[13] X. Huang, Z. Hu, Y. Qiao, and S. Sukkarieh, "Deep learning-based cow tail detection and tracking for precision livestock farming," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 3, pp. 1213–1221, 2022.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[15] G. Jocher, A. Chaurasia, and J. Qiu, "Yolo by ultralytics," https://github.com/ultralytics/ultralytics, Jan. 2023, version 8.0.0.

[16] G. Yang, J. Wang, Z. Nie, H. Yang, and S. Yu, "A lightweight yolov8 tomato detection algorithm combining feature enhancement and attention," *Agronomy*, vol. 13, no. 7, p. 1824, 2023.

[17] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[18] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," *CoRR*, vol. abs/1912.01703, 2019. [Online]. Available: http://arxiv.org/abs/1912.01703