BEHAVIOR RECOGNITION FOR ESTRUS MONITORING IN CATTLE COWS

INTRODUCTION

- Estrus detection is essential for cattle breeding effiency and farm productivity.
- Traditional observation is manual, time-consuming and inaccurate.
- Computer vision and deep learning can automate monitoring using video data.

PROJECT OBJECTIVE

OBJECTIVES

- Develop a model to classify cow behaviors (standing, lying, grazing & mounting) using video.
- Support estrus detection through automated behavior recognition,

PROJECT SCOPE

SCOPE

- Hardware: CCTV Camera, WiFi Access point (For now open-source dataset)
- Software: YOLO models, image processing & machine learning
- Output: Behavior recognition & visualization annotated data

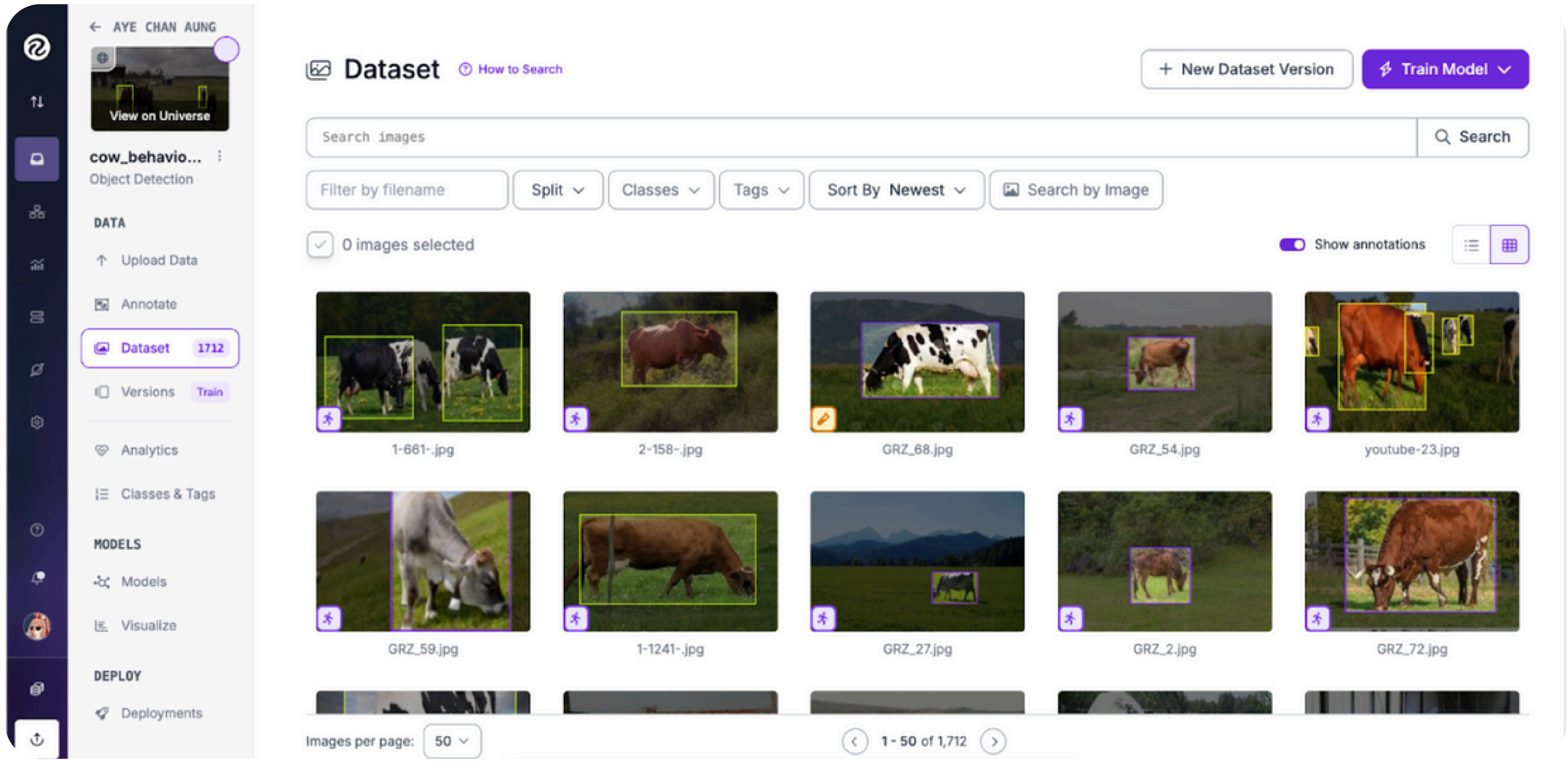
METHODOLOGY

This project focuses on building a cow posture object detection baseline.

The system takes images or video frames as input and outputs bounding boxes with one of four posture labels: grazing, lying, mounting, standing.

The workflow consists of five main stages:

1. Data collection from public sources
2. Annotation and dataset construction in Roboflow
3. Image preprocessing and augmentation
4. Model training with YOLOv8 and YOLOv11
5. Quantitative evaluation on a held-out test set
6. Qualitative testing on real farm CCTV footage

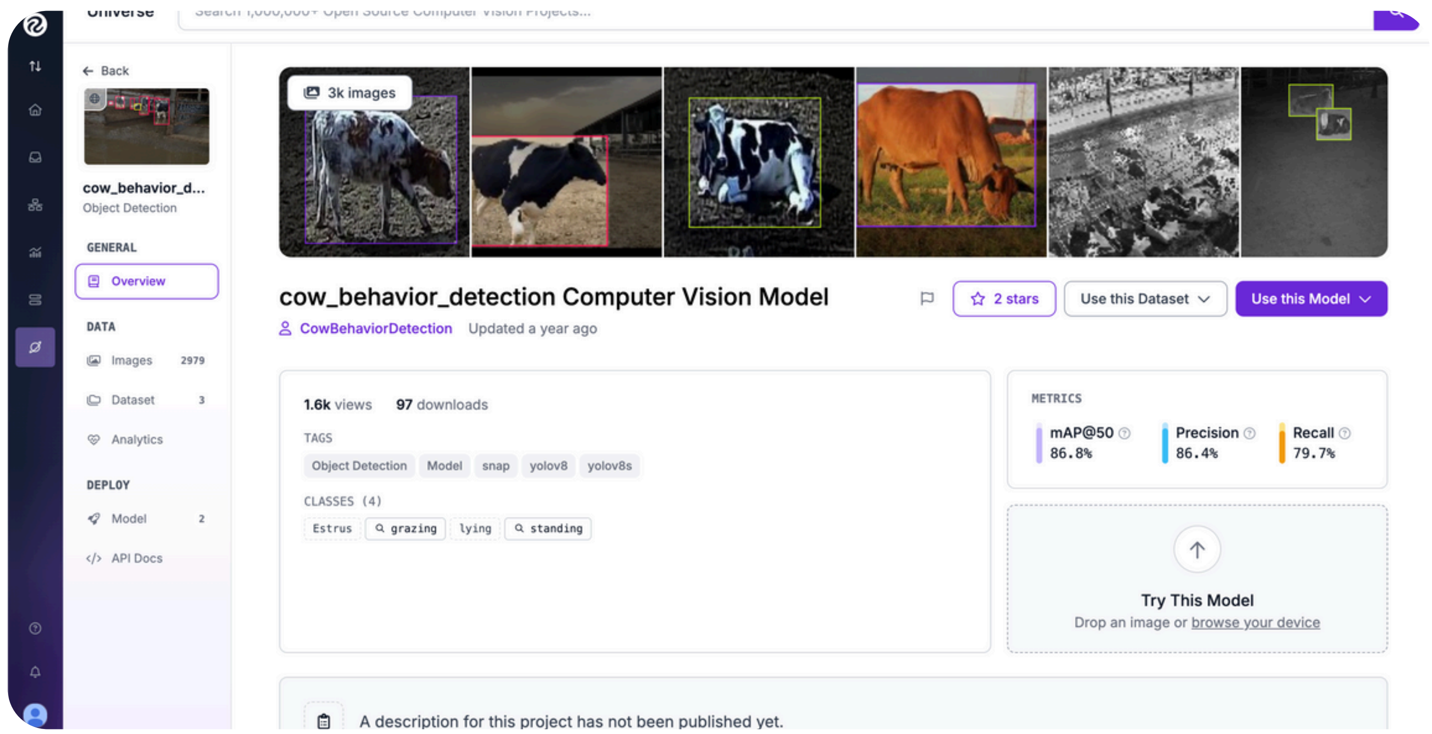


DATA COLLECTION

Images were obtained from two public sources:

- Roboflow Universe project – “Cow_Behavior_Detection”
- Web-scraped images using keywords such as “cow”, “cattle”, “cow farm”, “grazing cattle”, “lying cow”, and “mounting cow”

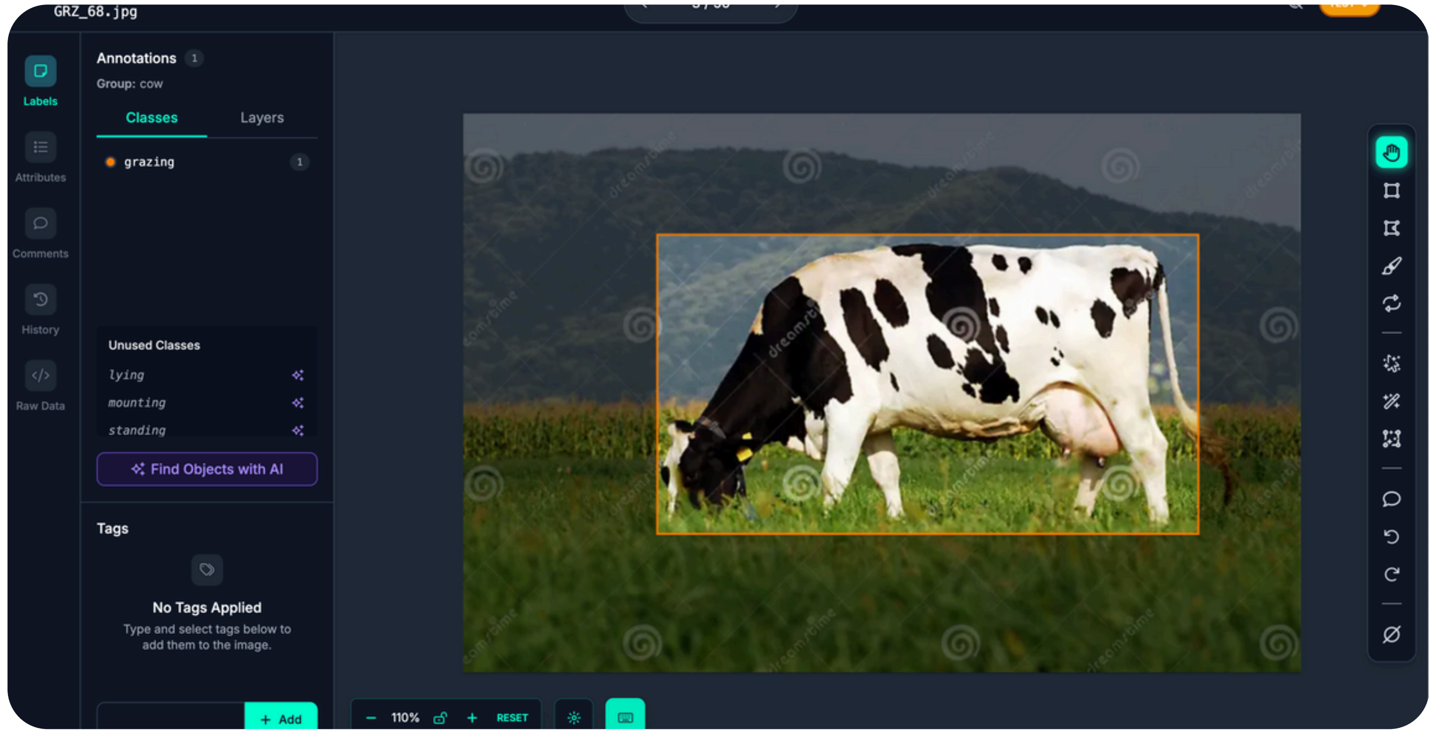
Only images where cows were clearly visible and posture was recognizable were kept. Duplicates, very low-resolution, heavily blurred, or extremely dark images were discarded. All selected images were uploaded into a single Roboflow project.



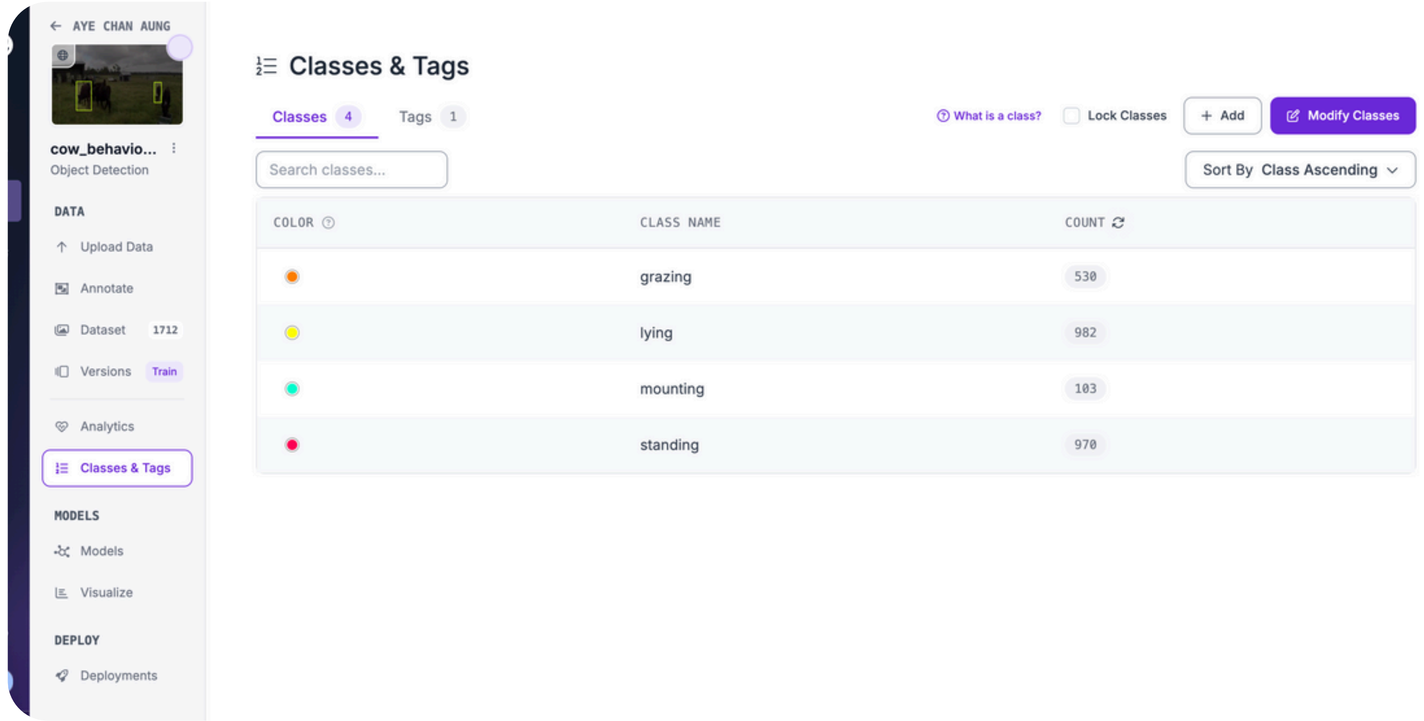
ANNOTATION AND DATASET CONSTRUCTION

Annotation was done in Roboflow using bounding boxes. Four posture classes were defined:

- grazing
- lying
- mounting
- standing



Every visible cow was annotated with one of these classes. The final dataset contains 1712 images with 530 grazing, 982 lying, 103 mounting, and 970 standing instances, revealing a strong class imbalance for mounting.



Roboflow exported the dataset in YOLO format (TXT labels + images). The data were randomly split into:

- 70% training set
- 15% validation set
- 15% test set

IMAGE PREPROCESSING AND AUGMENTATION

Before training, YOLO's built-in preprocessing pipeline is applied:

- Images are resized and letterboxed to 640 × 640 pixels (imgsz = 640).
- Pixel values are normalized to the range [0, 1] and arranged in 3-channel RGB.

During training we use extensive data augmentation to increase robustness and help control overfitting.

The main augmentation parameters are:

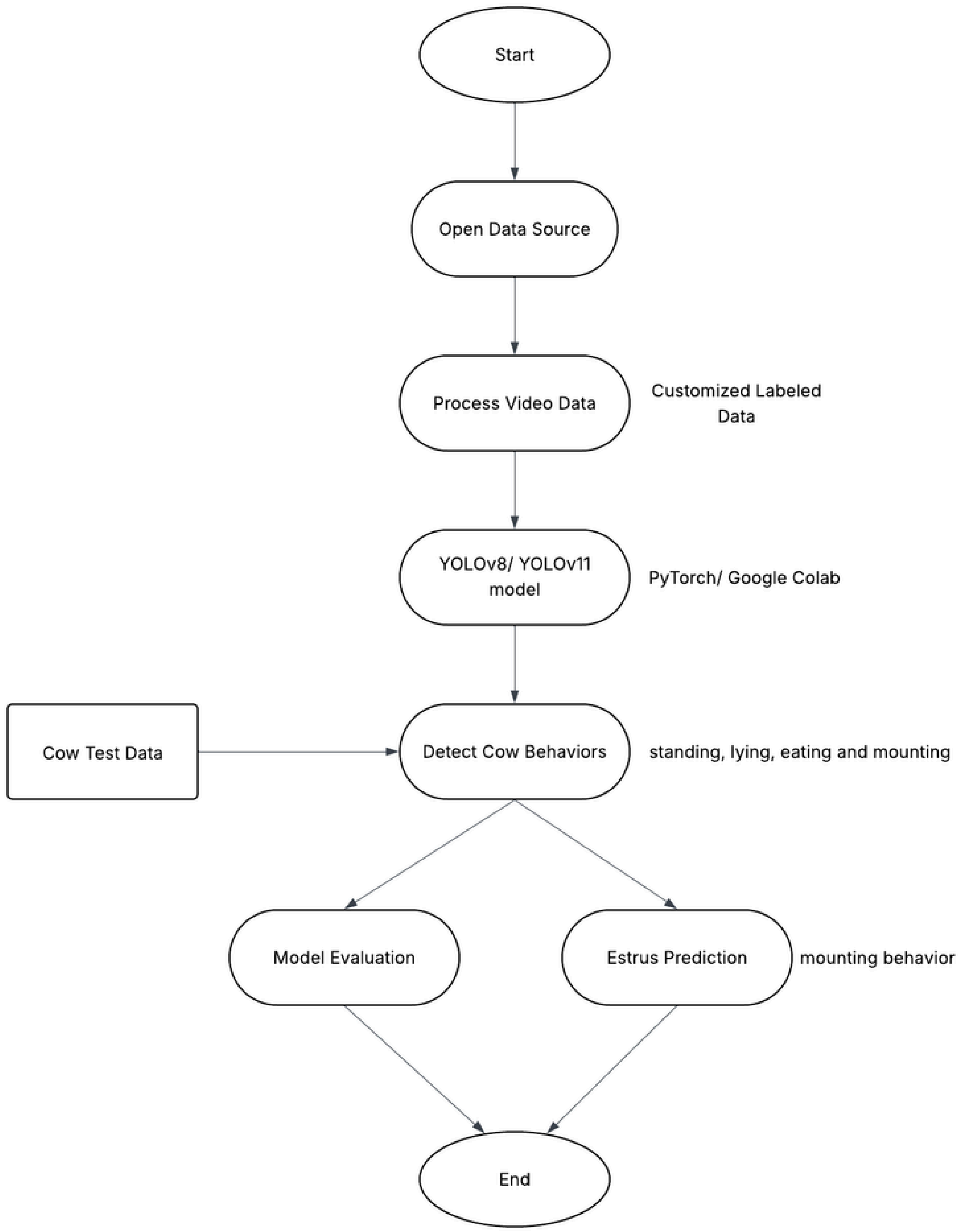
- Color jitter (HSV space):
  - hsv_h = 0.015, hsv_s = 0.7, hsv_v = 0.4
- Geometric transforms:
  - translate = 0.1 (±10% translation)
  - scale = 0.6 (scaling factor range)
  - degrees = 0.0, shear = 0.0, perspective = 0.0 (no rotation/shear in this setup)
- Flips:
  - vertical flip disabled: flipud = 0.0
  - horizontal flip enabled: fliplr = 0.5
- Advanced augmentations:
  - mosaic = 1.0
  - mixup = 0.1
  - copy_paste = 0.0

OVERALL RESULTS OF OBJECT DETECTION MODELS (TEST SET)

Model	Params (M)	Inference (ms/img)	~FPS	Training time (h)
YOLOv8s	11.1	4.4	227.3	0.359
YOLOv8n	3	**2.1**	**476.2**	0.332
YOLOv11n	2.6	3.2	312.5	0.347
YOLOv8m	25.8	10.3	97.1	0.64
YOLOv11s	9.4	4.8	208.3	0.384
YOLOv8l	43.6	15.4	64.9	1.043
YOLOv11m	20	10.5	95.2	0.639
YOLOv11l	25.3	13.5	74.1	1.019
RTDETR-M	32	16.3	61.3	2.079

Key point: YOLOv8s gives best accuracy, YOLOv8n is slightly less accurate but faster → both good candidates for real-time use.

SYSTEM FLOW (DIAGRAM)



System workflow from data collection to estrus prediction

TECHNOLOGY USED

OpenCV

PyTorch

Technologies

- YOLOv8/ YOLOv11 model - behavior detection
- Google Colab
- OpenCV + Pytorch - video processing & prediction

MODELS AND TRAINING

MODELS: YOLOV8 & YOLOV11

- YOLOv8 and YOLOv11 →modern single-stage detectors with Ultralytics API. Cow Posture Detection in Cattle...
- Model scales: n, s, m, l
- n = nano (fastest, smallest)
- s = small (good balance)
- m, l = medium & large (higher capacity)
- YOLOv11 adds:
- Improved C3k2 blocks, attention mechanisms
- Refined SPPF-like module for better accuracy–efficiency

TRAINING ENVIRONMENT & INTEGRATION

- Initial training on Google Colab with NVIDIA Tesla T4 GPU.
- Later training & testing on local machine + VS Code, using Ultralytics YOLO API. Cow Posture Detection in Cattle...
- Best-performing models saved as .pt:
- Easy to deploy later on web service / edge device.
- Current prototype:
- Offline Python script: read images/videos → output bounding boxes + posture labels.

TRAINING HYPERPARAMETERS

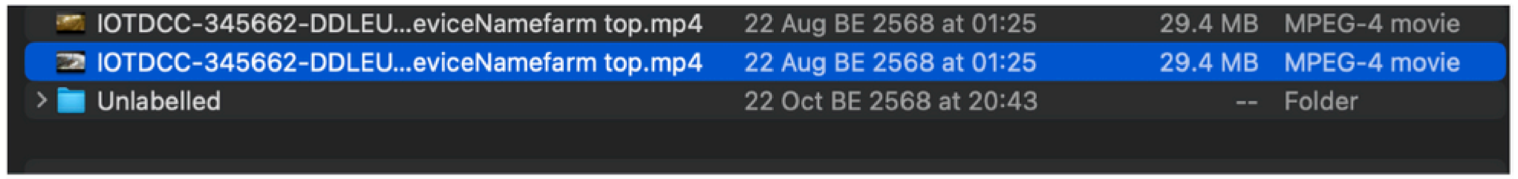
- Common settings for all YOLOv8/YOLOv11 variants (fair comparison): Cow Posture Detection in Cattle...
- Epochs: 50
- Batch size: 16, imgsz: 640
- Optimizer: SGD-like defaults
- lr0 = 0.001, lrf = 0.01
- Momentum = 0.937, Weight decay = 0.0005
- Warmup: 3 epochs, warmup momentum 0.8
- Early stopping with patience = 20 to reduce overfitting.

DATA AUGMENTATION

- Applied inside Ultralytics training pipeline: Cow Posture Detection in Cattle...
- HSV: h = 0.015, s = 0.7, v = 0.4
- Transform: translate = 0.1, scale = 0.6
- Flips: horizontal = 0.5
- Mosaic = 1.0, MixUp = 0.1
- Purpose:
- Increase robustness to lighting, background clutter, camera distance/orientation
- Reduce overfitting on relatively small dataset.

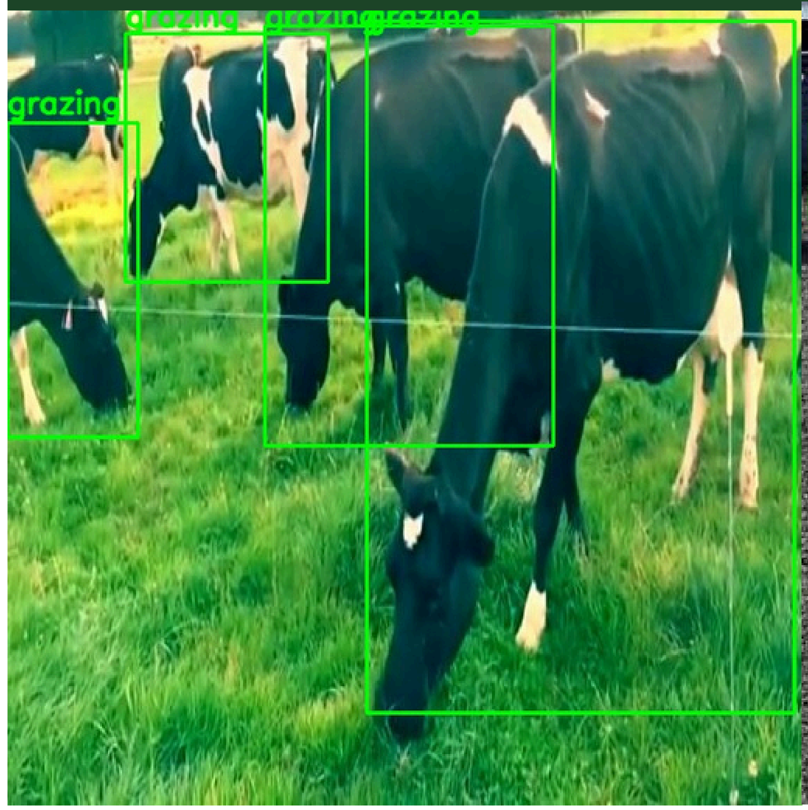
EXPERIMENT SETUP

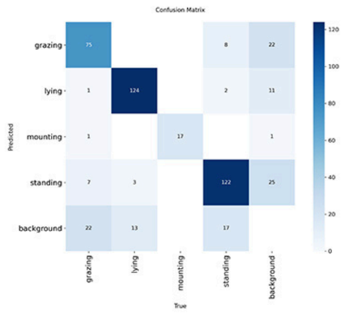
Example test using actual farm CCTV footage and YOLO detection model

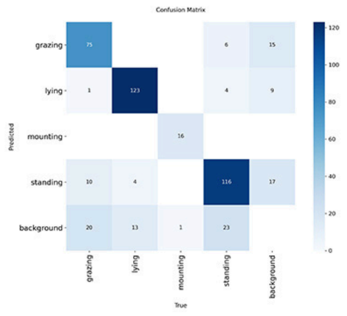


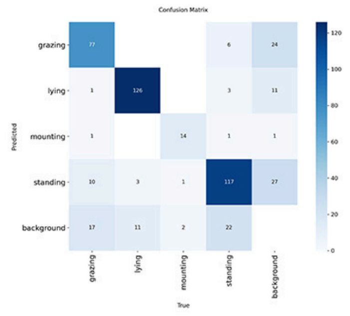
CONFUSION MATRIX & SAMPLE OUTPUTS

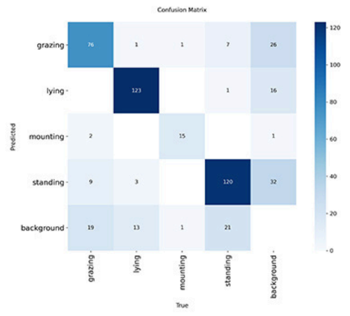
- Strong diagonal for standing & lying → consistently correct.
- Grazing sometimes misclassified as background under tall grass.
- Mounting occasionally confused with standing/grazing under occlusion.

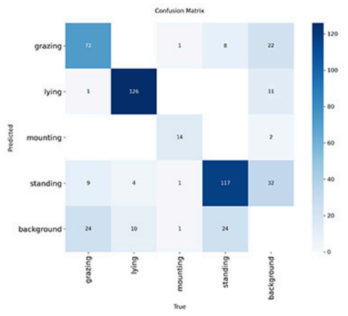


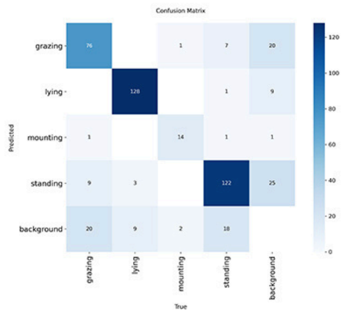

confusion matrix yolov8n

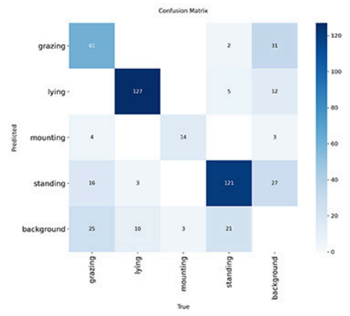

confusion matrix yolov8s

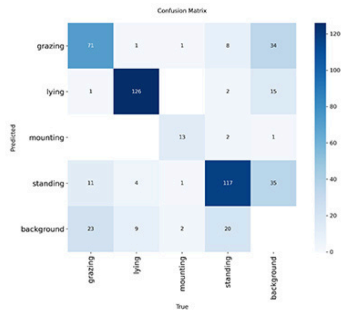

confusion matrix yolov8m


confusion matrix yolov8l


confusion matrix yolov11n


confusion matrix yolov11s


confusion matrix yolov11m


confusion matrix yolov11l

PER-CLASS PERFORMANCE – TOP 3 MODELS

YOLOv8s (best overall mAP)

Class	Precision	Recall	AP@50	AP@50–95
Overall	0.877	0.834	0.888	0.675
grazing	0.801	0.698	0.792	0.491
lying	0.909	0.886	0.922	0.761
mounting	**1**	**0.985**	**0.995**	**0.808**
standing	0.797	0.766	0.845	0.641

YOLOv8n (fastest, strong standing/mounting)

Class	Precision	Recall	AP@50	AP@50–95
Overall	0.873	0.824	0.866	0.667
grazing	0.809	0.689	0.755	0.479
lying	0.925	0.876	0.925	0.729
mounting	0.903	**0.941**	0.931	0.802
standing	0.855	0.792	0.851	**0.66**

YOLOv11s (best for grazing & lying)

Class	Precision	Recall	AP@50	AP@50–95
Overall	0.835	0.802	0.849	0.65
grazing	**0.801**	0.685	0.777	**0.503**
lying	**0.948**	**0.913**	**0.946**	**0.772**
mounting	0.792	0.824	0.818	0.668
standing	0.799	0.785	0.856	0.658

CHALLENGES:
OVERFITTING & UNDERFITTING

ISSUES FOUND WHEN TESTING ON REAL FARM CCTV:

- Differences from open-source images:
  - Lighting changes, occlusions, motion blur, varying camera angles
- Overfitting:
  - Good performance on training data, degraded on real farm videos
- Underfitting:
  - Model fails to learn some behaviors robustly

MITIGATION STRATEGIES

- Collect more real farm data in Northern Thailand
- Stronger augmentation (brightness, blur, rotation, scaling)
- Early stopping, regularization (L2, dropout)
- Adjust model size, learning rate, and epochs
- Integrate pose-based learning (DeepLabCut, etc.)

CONCLUSION

- Built a reproducible posture detection baseline using YOLOv8 & YOLOv11. Cow Posture Detection in Cattle...
- Achieved strong performance:
- mAP@50 up to 0.888 (YOLOv8s)
- ≈ 93% foreground micro accuracy
- Demonstrated that compact YOLO models can be used as:
- First step toward camera-only estrus monitoring
- Foundation for future behavior analytics and health anomaly detection.

FUTURE WORK

- Collect & label day/night CCTV data from multiple Thai farms. Cow Posture Detection in Cattle...
- Use DeepLabCut / Roboflow pose / SLEAP to get keypoints (head, back, legs, tail) for fine-grained posture.
- Add temporal models (temporal CNN, RNN, transformer) to:
- Detect mounting bursts
- Track long-term lying/grazing changes
- Integrate into a real-time pipeline with dashboard + alerts for estrus and health events.

THANK YOU

Thank you for supporting our project. We are very grateful for your advice and support.