# Pose Estimation for Livestock pose estimation

## Project Objective

## This project aims to develop an automated pose estimation system that will be used to identify and classify livestock (e.g., cattle) behaviors such as standing, lying, walking, and feeding from images. Pose estimation involves detecting keypoints of the animal body and analyzing their spatial locations, structures, and patterns to infer behaviors. The expected outcomes of the project will provide an enabling and accurate pose estimation system that is ready for monitoring the livestock health and welfare, which hence will support precision livestock farming. This project involves the work including (but not limited to) (1) Selecting a target livestock species (e.g., cattle, sheep); (2) Acquiring or annotating a dataset with keypoints; (3) Implementing pose estimation using open-source frameworks; (4) Evaluating model performance and model refinement; (5) Mapping poses to behavioural categories; (6) Deployment of a demo interface for real-time validation.

## Specific required knowledge, skills, and/or technology: 1) Python programming; (2) Basics of deep learning and computer vision; (3) use of image annotation tools

## Expected Outcomes

### Comprehensive Survey

**Current research on livestock behavior pose estimation(at least 5 research).**

A formal literature review document including:

1. Research target (e.g. Key points tracking, segmentation)
2. Analysis method (e.g. Deep Learning, statistic method)
3. Data acquisition (e.g. image, sensor data)
4. Metric (e.g. ROC, AUC, TPR)
5. Hypothesis/Prior Knowledge (e.g. limping and normal walking)
6. Hardware Requirements (e.g. GPU, memory)
7. Limitations (e.g. not support real-time inference)
8. Operational Efficiency(e.g. FPS, FLOPs)

### Experiment Preparation and Setting

**Select research targets based on your personnel and equipment situation**

**Prioritizing completion over performance.**

A document including:

1. Selecting Research Livestock: (sheep/cattle)
2. Computing resources (e.g. CPU num, GPU num)
3. Time Schedule (e.g. week2: literature review, week7: prototype demo)

A statistical report of dataset including:

1. The selected public dataset information(e.g. website, organization, source)
2. The statistical analysis (e.g. size of image, hyperspectral/natural image, SNR?)
3. Why choose this dataset?
4. How to use this dataset? (e.g. only use image, annotation is new.)
5. Train/Evaluation/Test division

A logic graph of livestock behavior estimation method

1. Defining three behaviors: (e.g. lying = cattle body touches the ground)
2. Methodology (e.g. first detect the cattle, then detect the ground)
3. Specific method for each sub target(e.g. detecting the cattle using CNN)
4. Visualization(e.g. how you show the detecting result?)
5. Target of real-time operational efficiency(e.g. FPS, FLOPs)

### End-to-end livestock behavior system

**Consider that your users are not computer professionals and only understand basic computer operations(Linux terminal is supported 😊).**

A User Guideline document including:

1. Manual: (e.g. How to install: pip install -e ., ./install.sh)
2. Hardware Requirements (e.g. CPU num, GPU num)
3. Function introduction (e.g. this system is designed for XXXXXX)
4. Limitations (e.g. this system can only XXXX, not XXXXX)
5. Frequently Asked Questions (e.g. why system is not responding? cd ~/main)
6. Future Study (e.g. XXXX will be online soon~)

A technical report including:

1. The biggest challenge (e.g. it is hard to detect the distance using image)
2. Advantages and disadvantages (e.g. good for adult cattles, hard for baby cattles)
3. Limitations and cause (e.g. Cattles being blocked can cause crashes, why?)
4. Future Study (e.g. Since poses estimation, how about mood based on poses?)

A system demo including:

1. Automatic system deployment (e.g. one-line terminal instruction)
2. Pre-loaded setting for detection (e.g. we recommend XXXX)
3. Training new data with customized settings(e.g. you can train you new data)
4. Detection new data with customized settings(e.g. specify the output to XXX )
5. Visualization

Requested function/file:

1. Dataloader (e.g. file I/O, annotation tools)
2. Data augmentation(e.g. flip, rotate)
3. Train(e.g. optimizer, scheduler)
4. Detection(e.g. predict -input XXX.rng -output ../result)
5. Visualization
6. Metric

## Expected Knowledge and Skills

### Prior Knowledge (Recommended)

* Python programming
* Basics of deep learning and computer vision
* Familiarity with image annotation tools

### Skills to Be Developed

* Pose estimation using deep learning
* Keypoint annotation and dataset handling
* Behaviour classification from spatial data
* Model evaluation and optimization
* Team collaboration and technical reporting

## Tools & Technologies

* Python, OpenCV
* Pose Estimation: OpenPose, MediaPipe, DeepLabCut
* ML Frameworks: PyTorch or TensorFlow
* Annotation Tools: CVAT, LabelMe
* GitHub for version control
* Optional: Streamlit or Flask for demo interface

## Public Datasets

### 1. CattlePoseEstimationDataset

Source: https://huggingface.co/datasets/gtsaidata/CattlePoseEstimationDataset/blob/main/README.md  
Description: Annotated with 12 keypoints (nose, eyes, hooves, tail, etc.) in COCO format.  
Use Case: Ideal for training pose estimation models like YOLOv8, DeepLabCut, HRNet.

### 2. CattleEyeView Dataset

Source: https://github.com/AnimalEyeQ/CattleEyeView  
Description: Top-down view cattle video dataset for multi-task learning (pose estimation, tracking, counting).  
Use Case: Useful for developing models that work with overhead surveillance footage.

## Open-Source Code Repositories

### CattleEyeView GitHub

https://github.com/AnimalEyeQ/CattleEyeView  
Includes dataset and benchmark code for pose estimation and other tasks.

### DeepLabCut

https://github.com/DeepLabCut/DeepLabCut  
Widely used for animal pose estimation with customizable keypoints.

### YOLOv8 Keypoint Detection

https://github.com/ultralytics/ultralytics  
Supports keypoint detection and compatible with COCO-format datasets.