**Lab 06 Task**

****

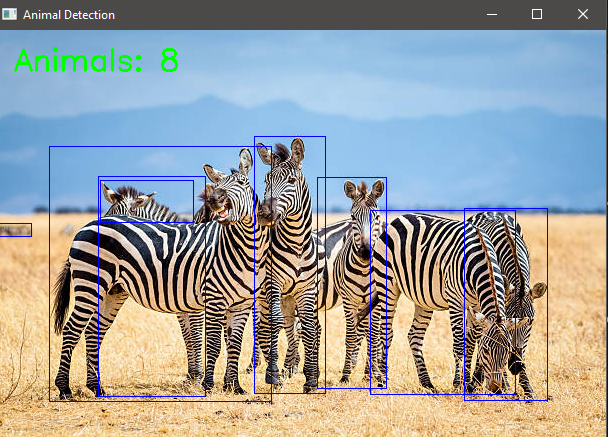
**Name Hammad Arshad**

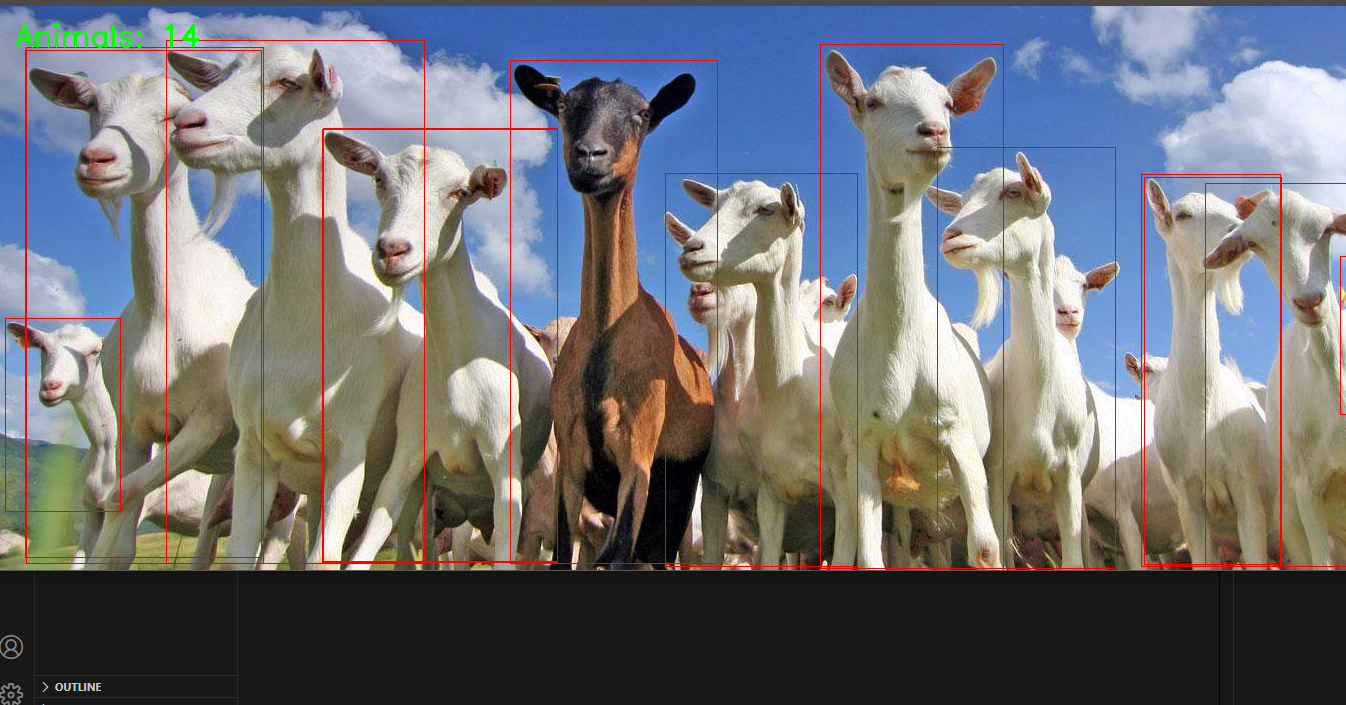
**Roll no SU92-BSAIM-F23-030**

**Section BSAI-4A**

**Subject PAI (LAB)**

**Animal Herd detection**





**HOW the Code Works**

**1. Libraries and Setup**

* **Ultralytics YOLO**: Provides the pre-trained object detection model
* **OpenCV (cv2)**: Handles image loading/display and drawing functions

**2. Model Initialization**

* model = YOLO("yolov8s.pt"): Loads the YOLOv8 small model, which balances speed and accuracy
* animal\_ids = [14, 15, 16, 17, 18, 19, 20, 21, 22, 23]: Filters for COCO dataset animal classes only:
  + These correspond to: bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, and giraffe

**3. Image Processing Pipeline**

* **Loading**: cv2.imread() reads the image file into a NumPy array
* **Detection**: model(img, classes=animal\_ids) runs inference on the image, filtering for animals only
* **Extraction**: .boxes retrieves all bounding box data from results
* **Visualization**:
  + Blue rectangles (255,0,0 in BGR) drawn around each detected animal
  + Green text showing total animal count added to top-left

**4. User Interaction**

* Prompts for input image path
* Displays results visually
* Waits for key press to close window

**WHY These Approaches Were Chosen**

**1. Why YOLO?**

* **Efficiency**: YOLO processes the entire image in one pass ("You Only Look Once")
* **Speed**: YOLOv8 can run in real-time on modern hardware
* **Accuracy**: The "s" (small) model balances computational requirements with detection quality

**2. Why Filter for Animal IDs?**

* **Focus**: Eliminates non-animal detections (people, vehicles, etc.)
* **Simplicity**: Makes the output cleaner by targeting only relevant classes
* **Efficiency**: Reduces unnecessary processing of irrelevant objects