

## ▼ Cow Pose YOLO Pipeline

This notebook trains, evaluates, compares, and exports YOLO Pose models with experiment tracking, TensorBoard logs, and inference benchmarks.

Key features:

- Experiment tracking under `(experiments/)` with per-run folders
- TensorBoard visualization
- Pose-specific augmentations
- Model comparison with mAP (pose), speed, and memory
- ONNX export and benchmarking

### ▼ 1. Install Dependencies

Run the cell below in Colab/Kaggle if needed. For local, skip if already installed.

```
1 # Colab / Kaggle
2 !pip install -q ultralytics tensorboard onnx onnxruntime
3 # If you want ONNX GPU in Kaggle/Colab (optional):
4 !pip install -q onnxruntime-gpu
```

1.2/1.2 MB 22.5 MB/s eta 0:00:00  
 17.5/17.5 MB 69.4 MB/s eta 0:00:00  
 17.1/17.1 MB 57.5 MB/s eta 0:00:00  
 252.6/252.6 MB 5.7 MB/s eta 0:00:00

### ▼ 2. Setup Paths

Adjust `(PROJECT_ROOT)` if you mounted Drive or copied the repo.

```
1 from google.colab import drive
2 drive.mount('/content/drive')

Mounted at /content/drive
```

```
1 !mkdir -p /content/cow_yolo_dataset
2 %cd /content/cow_yolo_dataset
3 !unzip -q /content/drive/MyDrive/cow_yolo_pose_dataset.zip
4 %cd /content
5 !unzip -q /content/drive/MyDrive/cow_pose_pipeline.zip
```

```
/content/cow_yolo_dataset
/content
```

```
1 import os
2 import sys
3 from pathlib import Path
4
5 PROJECT_ROOT = Path('.') # change if needed
6 sys.path.append(str(PROJECT_ROOT))
7
8 DATA_YAML = PROJECT_ROOT / 'cow_yolo_dataset' / 'dataset.yaml'
9 DATASET_DIR = PROJECT_ROOT / 'cow_yolo_dataset' # override if your dataset lives elsewhere
10
11 assert DATA_YAML.exists(), f'Dataset YAML not found at {DATA_YAML}'
12 assert DATASET_DIR.exists(), f'Dataset root not found at {DATASET_DIR}'
```

### ▼ 3. Configure Training

- `(model_id)`: YOLO pose medium (default: `(yolov8m-pose.pt)`)
- `(device)`: set `0` for single GPU, `[0, 1]` for multi-GPU

```
1 from cow_pose_pipeline import train_model, resolve_data_yaml
2
3 model_id = 'yolov8m-pose.pt'
4 epochs = 100
5 batch = -1 # auto
```

```

6 imgsz = 640
7 device = 0
8
9 resolved_yaml = resolve_data_yaml(DATA_YAML, dataset_dir=DATASET_DIR, out_dir=PROJECT_ROOT / 'experiments' /
10 resolved_yaml
11
PosixPath('experiments/configs/dataset_resolved.yaml')

```

## 4. Train (with Augmentation and Experiment Tracking)

Training outputs go to `experiments/pose/<timestep>/`.

```

1 weights = train_model(
2     data_yaml=resolved_yaml,
3     model_id=model_id,
4     epochs=epochs,
5     batch=batch,
6     imgsz=imgsz,
7     device=device,
8     project=PROJECT_ROOT / 'experiments',
9 )
10 weights

```

```

Downloading https://github.com/ultralytics/assets/releases/download/v8.4.0/yolov8m-pose.pt to 'yolov8m-pose.pt'
Ultralytics 8.4.12 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
engine/trainer: agnostic_nms=False, amp=True, angle=1.0, augment=False, auto_augment=randaugment, batch=16, bgr=
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf': 100% -
Overriding model.yaml kpt_shape=[17, 3] with kpt_shape=[26, 3]

```

	from	n	params	module	arguments
0	-1	1	1392	ultralytics.nn.modules.conv.Conv	[3, 48, 3, 2]
1	-1	1	41664	ultralytics.nn.modules.conv.Conv	[48, 96, 3, 2]
2	-1	2	111360	ultralytics.nn.modules.block.C2f	[96, 96, 2, True]
3	-1	1	166272	ultralytics.nn.modules.conv.Conv	[96, 192, 3, 2]
4	-1	4	813312	ultralytics.nn.modules.block.C2f	[192, 192, 4, True]
5	-1	1	664320	ultralytics.nn.modules.conv.Conv	[192, 384, 3, 2]
6	-1	4	3248640	ultralytics.nn.modules.block.C2f	[384, 384, 4, True]
7	-1	1	1991808	ultralytics.nn.modules.conv.Conv	[384, 576, 3, 2]
8	-1	2	3985920	ultralytics.nn.modules.block.C2f	[576, 576, 2, True]
9	-1	1	831168	ultralytics.nn.modules.block.SPPF	[576, 576, 5]
10	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
12	-1	2	1993728	ultralytics.nn.modules.block.C2f	[960, 384, 2]
13	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
15	-1	2	517632	ultralytics.nn.modules.block.C2f	[576, 192, 2]
16	-1	1	332160	ultralytics.nn.modules.conv.Conv	[192, 192, 3, 2]
17	[-1, 12]	1	0	ultralytics.nn.modules.conv.Concat	[1]
18	-1	2	1846272	ultralytics.nn.modules.block.C2f	[576, 384, 2]
19	-1	1	1327872	ultralytics.nn.modules.conv.Conv	[384, 384, 3, 2]
20	[-1, 9]	1	0	ultralytics.nn.modules.conv.Concat	[1]
21	-1	2	4207104	ultralytics.nn.modules.block.C2f	[960, 576, 2]
22	[15, 18, 21]	1	4768669	ultralytics.nn.modules.head.Pose	[1, [26, 3], 16, None, [192,

YOLOv8m-pose summary: 185 layers, 26,849,293 parameters, 26,849,277 gradients, 83.0 GFLOPs

```

Transferred 481/517 items from pretrained weights
Freezing layer 'model.22.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks...
Downloading https://github.com/ultralytics/assets/releases/download/v8.4.0/yolo26n.pt to 'yolo26n.pt': 100% -
AMP: checks passed ✅
train: Fast image access ✅ (ping: 0.0±0.0 ms, read: 3248.7±1015.4 MB/s, size: 179.4 KB)
train: Scanning /content/cow_yolo_dataset/labels/train... 226 images, 0 backgrounds, 0 corrupt: 100%
train: New cache created: /content/cow_yolo_dataset/labels/train.cache
WARNING⚠ No 'flip_idx' array defined in data.yaml, disabling 'fliplr' and 'flipud' augmentations.
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01, method='l')
val: Fast image access ✅ (ping: 0.0±0.0 ms, read: 928.0±572.5 MB/s, size: 216.9 KB)
val: Scanning /content/cow_yolo_dataset/labels/val... 25 images, 0 backgrounds, 0 corrupt: 100% —————— 25,
val: New cache created: /content/cow_yolo_dataset/labels/val.cache
Plotting labels to /content/runs/pose/experiments/pose/20260206_004652/labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', ''
optimizer: AdamW(lr=0.002, momentum=0.9) with parameter groups 83 weight(decay=0.0), 93 weight(decay=0.0005), 9:
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to /content/runs/pose/experiments/pose/20260206_004652
Starting training for 100 epochs...

```

Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size
1/100	6.6G	2.818	11.34	0.7088	5.363	3.41	2	640: 100%
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	Pose(P
	all	25	25	0.425	0.16	0.139	0.0411	R

## ▼ 5. TensorBoard

```
1 # !ps -aux | grep tensorboard
2 !pkill tensorboard
3 !ps -aux | grep tensorboard

root      14904  0.0  0.0    7376  3536 ?        S     01:28   0:00 /bin/bash -c ps -aux | grep tensorboard
root      14906  0.0  0.0    6484  2532 ?        S     01:28   0:00 grep tensorboard
```

```
1 # %reload_ext tensorboard
2 # %tensorboard --logdir /content/runs/pose/experiments/pose/20260206_004652 --bind_all
```

## ▼ 6. Evaluate + Benchmark (Pytorch)

This logs latency, FPS, and memory deltas.

```
1 from ultralytics import YOLO
2 from cow_pose_pipeline import benchmark_pytorch
3
4 weights = "/content/runs/pose/experiments/pose/20260206_004652/weights/best.pt"
5 model = YOLO(str(weights))
6 metrics = model.val(data=str(resolved_yaml), imgsz=imgsz, device=device, batch=1)
7 metrics
8
9 bench = benchmark_pytorch(
10     model=model,
11     data_yaml=resolved_yaml,
12     imgsz=imgsz,
13     device=device,
14     batch=1,
15 )
16 bench
17
```

```
Ultralytics 8.4.12 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
YOLOv8m-pose summary (fused): 102 layers, 26,832,265 parameters, 0 gradients, 82.6 GFLOPs
val: Fast image access ✓ (ping: 0.0±0.0 ms, read: 2117.7±431.2 MB/s, size: 175.3 KB)
val: Scanning /content/cow_yolo_dataset/labels/val.cache... 25 images, 0 backgrounds, 0 corrupt: 100%


| Class | Images | Instances | Box(P) | R   | mAP50 | mAP50-95 | Pose(P) | R |
|-------|--------|-----------|--------|-----|-------|----------|---------|---|
| all   | 25     | 25        | 0.619  | 0.4 | 0.383 | 0.159    | 0       | 0 |


Speed: 1.9ms preprocess, 49.8ms inference, 0.0ms loss, 2.1ms postprocess per image
Results saved to /content/runs/pose/val
-----  

RuntimeError                                     Traceback (most recent call last)
/tmp/ipython-input-4120309119.py in <cell line: 0>()
      7 metrics
      8
----> 9 bench = benchmark_pytorch(
     10     model=model,
     11     data_yaml=resolved_yaml,
```

▼ 18 frames ▼

```
/usr/local/lib/python3.12/dist-packages/torch/nn/modules/conv.py in _conv_forward(self, input, weight, bias)
  541         self.groups,
  542         )
-> 543     return F.conv2d(
  544         input, weight, bias, self.stride, self.padding, self.dilation, self.groups
  545     )
```

**RuntimeError:** Inference tensors do not track version counter.

```
1 from ultralytics import YOLO
2
3 weights = "/content/runs/pose/experiments/pose/20260206_004652/weights/best.pt"
4
5 # First validation
6 model = YOLO(str(weights))
7 metrics = model.val(data=str(resolved_yaml), imgsz=imgsz, device=device, batch=1)
8 print(metrics)
9
10 # RELOAD the model for benchmarking
11 model = YOLO(str(weights)) # Fresh load!
12
13 bench = benchmark_pytorch(
14     model=model,
15     data_yaml=resolved_yaml,
```

```
16     imgsz=imgsz,  
17     device=device,  
18     batch=1,  
19 )  
20 print(bench)
```

```
1 from ultralytics import YOLO
2 from ultralytics.utils.benchmarks import benchmark
3
4 weights = "/content/runs/pose/experiments/pose/20260206_004652/weights/best.pt"
5
6 # Use Ultralytics built-in benchmark
7 results = benchmark(
8     model=weights,
9     data=str(resolved_yaml),
10    imgsz=imgsz,
11    device=device,
12 )
13 print(results)
```

Setup complete ✓ (2 CPUs, 12.7 GB RAM, 47.3/112.6 GB disk)

Benchmarks complete for /content/runs/pose/experiments/pose/20260206\_004652/weights/best.pt on experiments/config  
Benchmarks legend: - ✓ Success - ✗ Export passed but validation failed - ✖ Export failed

Benchmarks legend:							
	Format	Status	Size (MB)	metrics/mAP50-95(P)	Inference time (ms/im)	FPS	
1	PyTorch	✓	51.5	0.0	20.84	47.99	
2	TorchScript	✓	102.9	0.0	24.72	40.44	

3	ONNX	✓	102.7	0.0	28.81	34.71
4	OpenVINO	✗	0.0	-	-	-
5	TensorRT	✓	138.4	0.0	21.39	46.75
6	CoreML	✗	0.0	-	-	-
7	TensorFlow SavedModel	✓	256.8	0.0	61.14	16.35
8	TensorFlow GraphDef	✗	102.7	-	-	-
9	TensorFlow Lite	✗	0.0	-	-	-
10	TensorFlow Edge TPU	✗	0.0	-	-	-
11	TensorFlow.js	✗	0.0	-	-	-
12	PaddlePaddle	✗	0.0	-	-	-
13	MNN	✓	102.6	0.0	972.95	1.03
14	NCNN	✓	102.6	0.0	1095.67	0.91
15	IMX	✗	0.0	-	-	-
16	RKNN	✗	0.0	-	-	-
17	ExecuTorch	✗	0.0	-	-	-
18	Axelera	✗	0.0	-	-	-

Format	Status ?	Size (MB)	metrics/mAP50-95(P)	Inference time (ms/im)	FPS
1 PyTorch	✓	51.5	0.0	20.84	47.99
2 TorchScript	✓	102.9	0.0	24.72	40.44
3 ONNX	✓	102.7	0.0	28.81	34.71
4 OpenVINO	✗	0.0	-	-	-
5 TensorRT	✓	138.4	0.0	21.39	46.75
6 CoreML	✗	0.0	-	-	-
7 TensorFlow SavedModel	✓	256.8	0.0	61.14	16.35
8 TensorFlow GraphDef	✗	102.7	-	-	-
9 TensorFlow Lite	✗	0.0	-	-	-
10 TensorFlow Edge TPU	✗	0.0	-	-	-
11 TensorFlow.js	✗	0.0	-	-	-
12 PaddlePaddle	✗	0.0	-	-	-
13 MNN	✓	102.6	0.0	972.95	1.03
14 NCNN	✓	102.6	0.0	1095.67	0.91
15 IMX	✗	0.0	-	-	-
16 RKNN	✗	0.0	-	-	-
17 ExecuTorch	✗	0.0	-	-	-
18 Axelera	✗	0.0	-	-	-

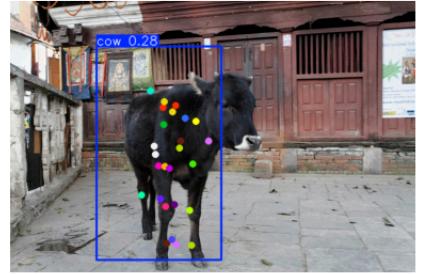
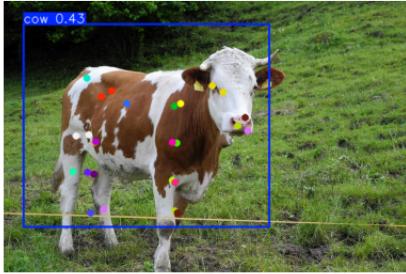
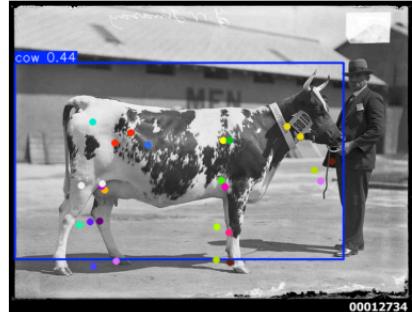
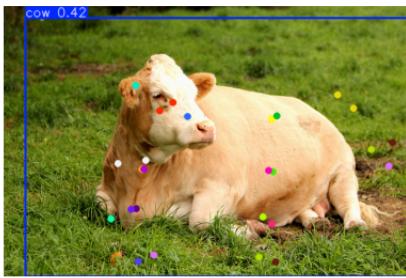
## ▼ 6b. Visualize Predictions (Val Set)

Render a few validation images with predicted keypoints and boxes.

```

1 import random
2 from pathlib import Path
3 import matplotlib.pyplot as plt
4 from ultralytics import YOLO
5
6 # Load model
7 viz_model = YOLO(str(weights))
8
9 # Pick a few validation images
10 val_dir = Path(DATASET_DIR) / 'images' / 'val'
11 val_images = list(val_dir.rglob('*.*')) + list(val_dir.rglob('*.png'))
12 sample = random.sample(val_images, k=min(6, len(val_images)))
13
14 results = viz_model.predict(source=sample, imgsz=imgsz, device=device, verbose=False)
15
16 # Plot
17 plt.figure(figsize=(12, 8))
18 for i, r in enumerate(results):
19     annotated = r.plot() # BGR image with boxes + keypoints
20     plt.subplot(2, 3, i + 1)
21     plt.imshow(annotated[:, :, ::-1])
22     plt.axis('off')
23 plt.tight_layout()
24

```



Accuracy is super low when i trained for 100 epochs.

While looking at the training\_batch image i noticed something really weird,  
Bounding boxes were incorrect. Thats explains the mAP being zero.

```
1 from IPython.display import display, Image  
2 Image("/content/drive/MyDrive/train_batch0.jpg")
```



```
1 !mkdir "/content/drive/MyDrive/cow_pose_training"
2 %cd "/content/"
```

```
/content
```

```
1 from cow_pose_pipeline import train_model, resolve_data_yaml
2
3 model_id = 'yolov8m-pose.pt'
4 epochs = 100
5 batch = -1 # auto
6 imgsz = 640
7 device = 0
8
9 resolved_yaml = resolve_data_yaml(DATA_YAML, dataset_dir=DATASET_DIR, out_dir=PROJECT_ROOT / 'experiments')
10 resolved_yaml
11
12 weights = train_model(
13     data_yaml=resolved_yaml,
14     model_id=model_id,
15     epochs=epochs,
16     batch=batch,
17     imgsz=imgsz,
18     device=device,
19     project="/content/drive/MyDrive/cow_pose_training/experiments",
20 )
21 weights
```

```
Downloading https://github.com/ultralytics/assets/releases/download/v8.4.0/yolov8m-pose.pt to 'yolov8m-pose.pt'
Ultralytics 8.4.12 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
```



```
ap_class_index: array([0])
box: ultralytics.utils.metrics.Metric object
confusion_matrix: <ultralytics.utils.metrics.ConfusionMatrix object at 0x7faa98d7e7b0>
curves: ['Precision-Recall(B)', 'F1-Confidence(B)', 'Precision-Confidence(B)', 'Recall-Confidence(B)', 'Precision-Recall(A)']
curves_results: [[array([
    0, 0.001001, 0.002002, 0.003003, 0.004004, 0.005005, 0.006006,
    0.024024, 0.025025, 0.026026, 0.027027, 0.028028, 0.029029, 0.03003,
    0.048048, 0.049049, 0.05005, 0.051051, 0.052052, 0.053053, 0.054054,
    0.072072, 0.073073, 0.074074, 0.075075, 0.076076, 0.077077, 0.078078,
    0.096096, 0.097097, 0.098098, 0.099099, 0.1001, 0.1011, 0.1021,
    0.12012, 0.12112, 0.12212, 0.12312, 0.12412, 0.12513, 0.12613,
    0.14414, 0.14515, 0.14615, 0.14715, 0.14815, 0.14915, 0.15015,
    0.16817, 0.16917, 0.17017, 0.17117, 0.17217, 0.17317, 0.17417,
    0.19219, 0.19319, 0.19419, 0.1952, 0.1962, 0.1972, 0.1982,
    0.21622, 0.21722, 0.21822, 0.21922, 0.22022, 0.22122, 0.22222,
    0.24024, 0.24124, 0.24224, 0.24324, 0.24424, 0.24525, 0.24625,
    0.26426, 0.26527, 0.26627, 0.26727, 0.26827, 0.26927, 0.27027,
    0.28829, 0.28929, 0.29029, 0.29129, 0.29229, 0.29329, 0.29429,
    0.31231, 0.31331, 0.31431, 0.31532, 0.31632, 0.31732, 0.31832,
    0.33634, 0.33734, 0.33834, 0.33934, 0.34034, 0.34134, 0.34234,
    0.36036, 0.36136, 0.36236, 0.36336, 0.36436, 0.36537, 0.36637,
    0.38438, 0.38539, 0.38639, 0.38739, 0.38839, 0.38939, 0.39039,
    0.40841, 0.40941, 0.41041, 0.41141, 0.41241, 0.41341, 0.41441,
    0.43243, 0.43343, 0.43443, 0.43544, 0.43644, 0.43744, 0.43844,
    0.45646, 0.45746, 0.45846, 0.45946, 0.46046, 0.46146, 0.46246,
    0.48048, 0.48148, 0.48248, 0.48348, 0.48448, 0.48549, 0.48649,
    0.5045, 0.50551, 0.50651, 0.50751, 0.50851, 0.50951, 0.51051,
    0.52853, 0.52953, 0.53053, 0.53153, 0.53253, 0.53353, 0.53453,
    0.55255, 0.55355, 0.55455, 0.55556, 0.55656, 0.55756, 0.55856,
    0.57658, 0.57758, 0.57858, 0.57958, 0.58058, 0.58158, 0.58258,
    0.6006, 0.6016, 0.6026, 0.6036, 0.6046, 0.60561, 0.60661,
    0.62462, 0.62563, 0.62663, 0.62763, 0.62863, 0.62963, 0.63063,
    0.64865, 0.64965, 0.65065, 0.65165, 0.65265, 0.65365, 0.65465,
    0.67267, 0.67367, 0.67467, 0.67568, 0.67668, 0.67768, 0.67868,
    0.6967, 0.6977, 0.6987, 0.6997, 0.7007, 0.7017, 0.7027,
    0.72072, 0.72172, 0.72272, 0.72372, 0.72472, 0.72573, 0.72673,
    0.74474, 0.74575, 0.74675, 0.74775, 0.74875, 0.74975, 0.75075,
    0.76877, 0.76977, 0.77077, 0.77177, 0.77277, 0.77377, 0.77477,
    0.79279, 0.79379, 0.79479, 0.7958, 0.7968, 0.7978, 0.7988,
    0.81682, 0.81782, 0.81882, 0.81982, 0.82082, 0.82182, 0.82282,
    0.84084, 0.84184, 0.84284, 0.84384, 0.84484, 0.84585, 0.84685,
    0.86486, 0.86587, 0.86687, 0.86787, 0.86887, 0.86987, 0.87087,
    0.88889, 0.88989, 0.89089, 0.89189, 0.89289, 0.89389, 0.89489,
    0.91291, 0.91391, 0.91491, 0.91592, 0.91692, 0.91792, 0.91892,
    0.93694, 0.93794, 0.93894, 0.93994, 0.94094, 0.94194, 0.94294,
    0.96096, 0.96196, 0.96296, 0.96396, 0.96496, 0.96597, 0.96697,
    0.98498, 0.98599, 0.98699, 0.98799, 0.98899, 0.98999, 0.99099,
    1., 1., 1., 1., 1., 1., 1., 1.])]
```

```
1 import random
2 from pathlib import Path
3 import matplotlib.pyplot as plt
4 from ultralytics import YOLO
5
6 # Load model
7 viz_model = YOLO(str(weights))
8
9 # Pick a few validation images
10 val_dir = Path(DATASET_DIR) / 'images' / 'val'
11 val_images = list(val_dir.rglob('*jpg')) + list(val_dir.rglob('*png'))
12 sample = random.sample(val_images, k=min(6, len(val_images)))
13
14 results = viz_model.predict(source=sample, imgsz=imgsz, device=device, verbose=False)
15
16 # Plot
17 plt.figure(figsize=(12, 8))
18 for i, r in enumerate(results):
19     annotated = r.plot() # BGR image with boxes + keypoints
20     plt.subplot(2, 3, i + 1)
21     plt.imshow(annotated[:, :, ::-1])
22     plt.axis('off')
23 plt.tight_layout()
24
```



▼ Accuracy still not good.

▼ Trying to fix accuracy.

```

1 # Run this to create a proper dataset.yaml
2
3 from pathlib import Path
4
5 dataset_root = Path("/content/cow_yolo_dataset")
6
7 yaml_content = """# YOLO Pose Dataset Configuration for Cow Pose Estimation
8 # FIXED with proper keypoint configuration
9
10 path: {path}
11 train: images/train
12 val: images/val
13
14 # Keypoint shape
15 kpt_shape: [26, 3]
16
17 # Classes
18 names:
19   0: cow
20
21 # =====
22 # CRITICAL: OKS Sigmas for evaluation
23 # =====
24 # These control how strictly keypoints are matched
25 # Smaller = stricter, Larger = more forgiving
26 # Range typically 0.025 (precise) to 0.10 (loose)
27
28 kpt_sigmas:
29   - 0.026 # 0 L_Eye - precise facial feature
30   - 0.026 # 1 R_Eye
31   - 0.035 # 2 Chin
32   - 0.055 # 3 R_F_Hoof - hooves can vary
33   - 0.055 # 4 L_F_Hoof

```

```

34 - 0.055 # 5 R_B_Hoof
35 - 0.055 # 6 L_B_Hoof
36 - 0.065 # 7 Tail_Base - larger body part
37 - 0.045 # 8 R_Shoulder
38 - 0.045 # 9 L_Shoulder
39 - 0.055 # 10 R_Hip
40 - 0.055 # 11 L_Hip
41 - 0.055 # 12 Spine_Mid
42 - 0.045 # 13 Withers
43 - 0.045 # 14 R_F_Knee
44 - 0.045 # 15 L_F_Knee
45 - 0.045 # 16 R_B_Knee
46 - 0.045 # 17 L_B_Knee
47 - 0.040 # 18 Throat
48 - 0.085 # 19 Tail_Tip - high variance
49 - 0.035 # 20 L_EarBase
50 - 0.035 # 21 R_EarBase
51 - 0.035 # 22 Mouth_Corner
52 - 0.030 # 23 Nose_Tip
53 - 0.030 # 24 Nose_Bridge
54 - 0.070 # 25 Tail_Mid
55
56 # =====
57 # CRITICAL: Flip indices for augmentation
58 # =====
59 # Maps each keypoint to its left-right counterpart
60 # Center keypoints map to themselves
61
62 flip_idx:
63 - 1 # 0 L_Eye ↔ R_Eye
64 - 0 # 1 R_Eye ↔ L_Eye
65 - 2 # 2 Chin (center)
66 - 4 # 3 R_F_Hoof ↔ L_F_Hoof
67 - 3 # 4 L_F_Hoof ↔ R_F_Hoof
68 - 6 # 5 R_B_Hoof ↔ L_B_Hoof
69 - 5 # 6 L_B_Hoof ↔ R_B_Hoof
70 - 7 # 7 Tail_Base (center)
71 - 9 # 8 R_Shoulder ↔ L_Shoulder
72 - 8 # 9 L_Shoulder ↔ R_Shoulder
73 - 11 # 10 R_Hip ↔ L_Hip
74 - 10 # 11 L_Hip ↔ R_Hip
75 - 12 # 12 Spine_Mid (center)
76 - 13 # 13 Withers (center)
77 - 15 # 14 R_F_Knee ↔ L_F_Knee
78 - 14 # 15 L_F_Knee ↔ R_F_Knee
79 - 17 # 16 R_B_Knee ↔ L_B_Knee
80 - 16 # 17 L_B_Knee ↔ R_B_Knee
81 - 18 # 18 Throat (center)
82 - 19 # 19 Tail_Tip (center)
83 - 21 # 20 L_EarBase ↔ R_EarBase
84 - 20 # 21 R_EarBase ↔ L_EarBase
85 - 22 # 22 Mouth_Corner (center)
86 - 23 # 23 Nose_Tip (center)
87 - 24 # 24 Nose_Bridge (center)
88 - 25 # 25 Tail_Mid (center)
89
90 # Keypoint names (reference)
91 keypoint_names:
92 - L_Eye
93 - R_Eye
94 - Chin
95 - R_F_Hoof
96 - L_F_Hoof
97 - R_B_Hoof
98 - L_B_Hoof
99 - Tail_Base
100 - R_Shoulder
101 - L_Shoulder
102 - R_Hip
103 - L_Hip
104 - Spine_Mid
105 - Withers
106 - R_F_Knee
107 - L_F_Knee
108 - R_B_Knee
109 - L_B_Knee
110 - Throat
111 - Tail_Tip
112 - L_EarBase
113 - R_EarBase
114 - Mouth_Corner
115 - Nose_Tip

```

```

116     - Nose_Bridge
117     - Tail_Mid
118 """ .format(path=str(dataset_root.absolute()))
119
120 # Write the file
121 yaml_path = dataset_root / "dataset_fixed.yaml"
122 with open(yaml_path, 'w') as f:
123     f.write(yaml_content)
124
125 print(f"✓ Fixed dataset_fixed.yaml written to: {yaml_path}")

```

✓ Fixed dataset\_fixed.yaml written to: /content/cow\_yolo\_dataset/dataset\_fixed.yaml

```

1 import shutil
2 from pathlib import Path
3
4 # Remove old cache files
5 dataset_root = Path("/content/cow_yolo_dataset")
6 for cache_file in dataset_root.glob("*.cache"):
7     cache_file.unlink()
8     print(f"Removed: {cache_file}")
9
10 # Remove old cache in labels folders
11 for split in ["train", "val"]:
12     cache = dataset_root / "labels" / split / ".cache"
13     if cache.exists():
14         cache.unlink()
15     print(f"Removed: {cache}")
16
17 print("✓ Cache cleared!")

```

Removed: /content/cow\_yolo\_dataset/labels/val.cache  
 Removed: /content/cow\_yolo\_dataset/labels/train.cache  
 ✓ Cache cleared!

```

1 from ultralytics import YOLO
2
3 # Use a smaller model given the small dataset
4 model = YOLO("yolov8s-pose.pt") # Small model - better for 251 samples
5
6 results = model.train(
7     data="/content/cow_yolo_dataset/dataset_fixed.yaml",
8     epochs=150,
9     imgsz=640,
10    batch=32, # Auto batch
11    patience=30,
12    device=0,
13    project="/content/drive/MyDrive/cow_pose_training/experiments",
14    name="with_kpt_sigmas",
15
16    # Hyper-parameters
17    degrees=10.0,
18    translate=0.1,
19    scale=0.3, # Reduced
20    fliplr=0.5,
21    mosaic=0.5,
22    mixup=0.0, # Disable
23
24    # Regularization
25    dropout=0.1,
26
27    # Learning rate
28    lr0=0.01,
29    lrf=0.01,
30 )

```

Downloading <https://github.com/ultralytics/assets/releases/download/v8.4.0/yolov8s-pose.pt> to 'yolov8s-pose.pt'  
 Ultralytics 8.4.12 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)  
**engine/trainer:** agnostic\_nms=False, amp=True, angle=1.0, augment=False, auto\_augment=randaugment, batch=32, bgr=Overriding model.yaml kpt\_shape=[17, 3] with kpt\_shape=[26, 3]

	from	n	params	module	arguments
0		-1	1	928 ultralytics.nn.modules.conv.Conv	[3, 32, 3, 2]
1		-1	1	18560 ultralytics.nn.modules.conv.Conv	[32, 64, 3, 2]
2		-1	1	29056 ultralytics.nn.modules.block.C2f	[64, 64, 1, True]
3		-1	1	73984 ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
4		-1	2	197632 ultralytics.nn.modules.block.C2f	[128, 128, 2, True]
5		-1	1	295424 ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
6		-1	2	788480 ultralytics.nn.modules.block.C2f	[256, 256, 2, True]
7		-1	1	1180672 ultralytics.nn.modules.conv.Conv	[256, 512, 3, 2]
8		-1	1	1838080 ultralytics.nn.modules.block.C2f	[512, 512, 1, True]
9		-1	1	656896 ultralytics.nn.modules.block.SPPF	[512, 512, 5]
10		-1	1	0 torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']

```

11      [-1, 6] 1      0 ultralytics.nn.modules.conv.Concat [1]
12      -1 1 591360 ultralytics.nn.modules.block.C2f [768, 256, 1]
13      -1 1      0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
14      [-1, 4] 1      0 ultralytics.nn.modules.conv.Concat [1]
15      -1 1 148224 ultralytics.nn.modules.block.C2f [384, 128, 1]
16      -1 1 147712 ultralytics.nn.modules.conv.Conv [128, 128, 3, 2]
17      [-1, 12] 1      0 ultralytics.nn.modules.conv.Concat [1]
18      -1 1 493056 ultralytics.nn.modules.block.C2f [384, 256, 1]
19      -1 1 590336 ultralytics.nn.modules.conv.Conv [256, 256, 3, 2]
20      [-1, 9] 1      0 ultralytics.nn.modules.conv.Concat [1]
21      -1 1 1969152 ultralytics.nn.modules.block.C2f [768, 512, 1]
22      [15, 18, 21] 1 2929117 ultralytics.nn.modules.head.Pose [1, [26, 3], 16, None, [128,
YOLOv8s-pose summary: 145 layers, 11,948,669 parameters, 11,948,653 gradients, 31.7 GFLOPs

```

Transferred 361/397 items from pretrained weights

Freezing layer 'model.22.dfl.conv.weight'

**AMP:** running Automatic Mixed Precision (AMP) checks...

**AMP:** checks passed ✓

**train:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 2097.0±637.3 MB/s, size: 184.5 KB)

**train:** Scanning /content/cow\_yolo\_dataset/labels/train... 226 images, 0 backgrounds, 0 corrupt: 100% ━━━━━

**train:** New cache created: /content/cow\_yolo\_dataset/labels/train.cache

**albumentations:** Blur(p=0.01, blur\_limit=(3, 7)), MedianBlur(p=0.01, blur\_limit=(3, 7)), ToGray(p=0.01, method='

**val:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 909.8±1076.2 MB/s, size: 230.9 KB)

**val:** Scanning /content/cow\_yolo\_dataset/labels/val... 25 images, 0 backgrounds, 0 corrupt: 100% ━━━━━━ 25,

**val:** New cache created: /content/cow\_yolo\_dataset/labels/val.cache

Plotting labels to /content/drive/MyDrive/cow\_pose\_training/experiments/with\_kpt\_sigmas/labels.jpg...

**optimizer:** 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', ''

**optimizer:** AdamW(lr=0.002, momentum=0.9) with parameter groups 63 weight(decay=0.0), 73 weight(decay=0.0005), 7:

Image sizes 640 train, 640 val

Using 2 dataloader workers

Logging results to /content/drive/MyDrive/cow\_pose\_training/experiments/with\_kpt\_sigmas

Starting training for 150 epochs...

Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size
1/150	8.61G	3.205	10.31	0.7134	5.521	3.894	4	640: 100% ━
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	Pose(P
	all	25	25	0.0151	0.36	0.0142	0.00344	R

Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size
2/150	7.81G	1.835	10.8	0.6954	1.994	2.213	2	640: 100% ━

```

1 from ultralytics import YOLO
2 from cow_pose_pipeline import benchmark_pytorch
3 weights = Path("/content/drive/MyDrive/cow_pose_training/experiments/with_kpt_sigmas/weights/best.pt")
4
5 # First validation
6 model = YOLO(str(weights))
7 metrics = model.val(data=str(resolved_yaml), imgsz=imgsz, device=device, batch=1)
8 print(metrics)
9
10 # RELOAD the model for benchmarking
11 model = YOLO(str(weights)) # Fresh load!
12
13 bench = benchmark_pytorch(
14     model=model,
15     data_yaml=resolved_yaml,
16     imgsz=imgsz,
17     device=device,
18     batch=1,
19 )
20 print(bench)

```

Ultralytics 8.4.12 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)

YOLOv8s-pose summary (fused): 82 layers, 11,938,185 parameters, 0 gradients, 31.5 GFLOPs

**val:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 2534.8±529.2 MB/s, size: 200.0 KB)

**val:** Scanning /content/cow\_yolo\_dataset/labels/val.cache... 25 images, 0 backgrounds, 0 corrupt: 100% ━━━━━

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	Pose(P	R
all	25	25	0.994	0.88	0.934	0.761	0.539	0.44

Speed: 1.2ms preprocess, 33.0ms inference, 0.0ms loss, 1.3ms postprocess per image

Results saved to /content/runs/pose/val3

ultralytics.utils.metrics.PoseMetrics object with attributes:

```

ap_class_index: array([0])
box: ultralytics.utils.metrics.Metric object
confusion_matrix: <ultralytics.utils.metrics.ConfusionMatrix object at 0x7faa98cef1d0>
curves: ['Precision-Recall(B)', 'F1-Confidence(B)', 'Precision-Confidence(B)', 'Recall-Confidence(B)', 'Precision-Recall(AUC)', 'F1-Confidence(AUC)', 'Precision-Confidence(AUC)', 'Recall-Confidence(AUC)']
curves_results: [[array([
    0, 0.001001, 0.002002, 0.003003, 0.004004, 0.005005, 0.006006,
    0.024024, 0.025025, 0.026026, 0.027027, 0.028028, 0.029029, 0.03003,
    0.048048, 0.049049, 0.05005, 0.051051, 0.052052, 0.053053, 0.054054,
    0.072072, 0.073073, 0.074074, 0.075075, 0.076076, 0.077077, 0.078078,
    0.096096, 0.097097, 0.098098, 0.099099, 0.1001, 0.1011, 0.1021,
    0.12012, 0.12112, 0.12212, 0.12312, 0.12412, 0.12513, 0.12613,
    0.14414, 0.14515, 0.14615, 0.14715, 0.14815, 0.14915, 0.15015,
    0.16817, 0.16917, 0.17017, 0.17117, 0.17217, 0.17317, 0.17417,
    0.19219, 0.19319, 0.19419, 0.1952, 0.1962, 0.1972, 0.1982,
    0.21622, 0.21722, 0.21822, 0.21922, 0.22022, 0.22122, 0.22222,
    0.24024, 0.24124, 0.24224, 0.24324, 0.24424, 0.24525, 0.24625,
    0.24725
])]]]

```

0.26426,	0.26527,	0.26627,	0.26727,	0.26827,	0.26927,	0.27027,	0.27127,
0.28829,	0.28929,	0.29029,	0.29129,	0.29229,	0.29329,	0.29429,	0.2953,
0.31231,	0.31331,	0.31431,	0.31532,	0.31632,	0.31732,	0.31832,	0.31932,
0.33634,	0.33734,	0.33834,	0.33934,	0.34034,	0.34134,	0.34234,	0.34334,
0.36036,	0.36136,	0.36236,	0.36336,	0.36436,	0.36537,	0.36637,	0.36737,
0.38438,	0.38539,	0.38639,	0.38739,	0.38839,	0.38939,	0.39039,	0.39139,
0.40841,	0.40941,	0.41041,	0.41141,	0.41241,	0.41341,	0.41441,	0.41542,
0.43243,	0.43343,	0.43443,	0.43544,	0.43644,	0.43744,	0.43844,	0.43944,
0.45646,	0.45746,	0.45846,	0.45946,	0.46046,	0.46146,	0.46246,	0.46346,
0.48048,	0.48148,	0.48248,	0.48348,	0.48448,	0.48549,	0.48649,	0.48749,
0.5045,	0.50551,	0.50651,	0.50751,	0.50851,	0.50951,	0.51051,	0.51151,
0.52853,	0.52953,	0.53053,	0.53153,	0.53253,	0.53353,	0.53453,	0.53554,
0.55255,	0.55355,	0.55455,	0.55556,	0.55656,	0.55756,	0.55856,	0.55956,
0.57658,	0.57758,	0.57858,	0.57958,	0.58058,	0.58158,	0.58258,	0.58358,
0.6006,	0.6016,	0.6026,	0.6036,	0.6046,	0.60561,	0.60661,	0.60761,
0.62462,	0.62563,	0.62663,	0.62763,	0.62863,	0.62963,	0.63063,	0.63163,
0.64865,	0.64965,	0.65065,	0.65165,	0.65265,	0.65365,	0.65465,	0.65566,
0.67267,	0.67367,	0.67467,	0.67568,	0.67668,	0.67768,	0.67868,	0.67968,
0.6967,	0.6977,	0.6987,	0.6997,	0.7007,	0.7017,	0.7027,	0.7037,
0.72072,	0.72172,	0.72272,	0.72372,	0.72472,	0.72573,	0.72673,	0.72773,
0.74474,	0.74575,	0.74675,	0.74775,	0.74875,	0.74975,	0.75075,	0.75175,
0.76877,	0.76977,	0.77077,	0.77177,	0.77277,	0.77377,	0.77477,	0.77578,
0.79279,	0.79379,	0.79479,	0.7958,	0.7968,	0.7978,	0.7988,	0.7998,
0.81682,	0.81782,	0.81882,	0.81982,	0.82082,	0.82182,	0.82282,	0.82382,
0.84084,	0.84184,	0.84284,	0.84384,	0.84484,	0.84585,	0.84685,	0.84785,
0.86486,	0.86587,	0.86687,	0.86787,	0.86887,	0.86987,	0.87087,	0.87187,
0.88889,	0.88989,	0.89089,	0.89189,	0.89289,	0.89389,	0.89489,	0.8959,
0.91291,	0.91391,	0.91491,	0.91592,	0.91692,	0.91792,	0.91892,	0.91992,
0.93694,	0.93794,	0.93894,	0.93994,	0.94094,	0.94194,	0.94294,	0.94394,
0.96096,	0.96196,	0.96296,	0.96396,	0.96496,	0.96597,	0.96697,	0.96797,
0.98498,	0.98599,	0.98699,	0.98799,	0.98899,	0.98999,	0.99099,	0.99199,

```

1 from ultralytics import YOLO
2 from ultralytics.utils.benchmarks import benchmark
3
4 # Use Ultralytics built-in benchmark
5 results = benchmark(
6     model=weights,
7     data=str(resolved_yaml),
8     imgsz=imgsz,
9     device=device,
10 )
11 print(results)

```

Setup complete ✓ (2 CPUs, 12.7 GB RAM, 47.5/112.6 GB disk)

Benchmarks complete for /content/drive/MyDrive/cow\_pose\_training/experiments/with\_kpt\_sigmas/weights/best.pt on e  
Benchmarks legend: - ✓ Success - ✘ Export passed but validation failed - ✗ Export failed

Format	Status ?	Size (MB)	metrics/mAP50-95(P)	Inference time (ms/im)	FPS
1 PyTorch	✓	23.1	0.1447	13.13	76.18
2 TorchScript	✓	46.0	0.1115	13.37	74.79
3 ONNX	✓	45.9	0.1115	17.44	57.33
4 OpenVINO	✗	0.0	-	-	-
5 TensorRT	✓	55.7	0.1115	8.05	124.17
6 CoreML	✗	0.0	-	-	-
7 TensorFlow SavedModel	✓	114.7	0.1115	62.86	15.91
8 TensorFlow GraphDef	✗	45.9	-	-	-
9 TensorFlow Lite	✗	0.0	-	-	-
10 TensorFlow Edge TPU	✗	0.0	-	-	-
11 TensorFlow.js	✗	0.0	-	-	-
12 PaddlePaddle	✗	0.0	-	-	-
13 MNN	✓	45.7	0.1115	486.4	2.06
14 NCNN	✓	45.8	0.1115	545.14	1.83
15 IMX	✗	0.0	-	-	-
16 RKNN	✗	0.0	-	-	-
17 ExecuTorch	✗	0.0	-	-	-
18 Axelera	✗	0.0	-	-	-

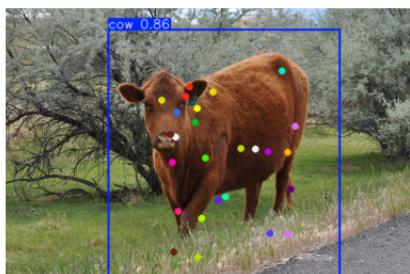
Format	Status ?	Size (MB)	metrics/mAP50-95(P)	Inference time (ms/im)	FPS
1 PyTorch	✓	23.1	0.1447	13.13	76.18
2 TorchScript	✓	46.0	0.1115	13.37	74.79
3 ONNX	✓	45.9	0.1115	17.44	57.33
4 OpenVINO	✗	0.0	-	-	-
5 TensorRT	✓	55.7	0.1115	8.05	124.17
6 CoreML	✗	0.0	-	-	-
7 TensorFlow SavedModel	✓	114.7	0.1115	62.86	15.91
8 TensorFlow GraphDef	✗	45.9	-	-	-
9 TensorFlow Lite	✗	0.0	-	-	-
10 TensorFlow Edge TPU	✗	0.0	-	-	-
11 TensorFlow.js	✗	0.0	-	-	-
12 PaddlePaddle	✗	0.0	-	-	-
13 MNN	✓	45.7	0.1115	486.4	2.06
14 NCNN	✓	45.8	0.1115	545.14	1.83

15	IMX	X	0.0	-	-	-	
16	RKNN	X	0.0	-	-	-	
17	ExecuTorch	X	0.0	-	-	-	
18	Axelera	X	0.0	-	-	-	

```

1 import random
2 from pathlib import Path
3 import matplotlib.pyplot as plt
4 from ultralytics import YOLO
5
6 weights = Path("/content/drive/MyDrive/cow_pose_training/experiments/with_kpt_sigmas/weights/best.pt")
7 # Load model
8 viz_model = YOLO(str(weights))
9
10 # Pick a few validation images
11 val_dir = Path(DATASET_DIR) / 'images' / 'val'
12 val_images = list(val_dir.rglob('*.*')) + list(val_dir.rglob('*.png'))
13 sample = random.sample(val_images, k=min(6, len(val_images)))
14
15 results = viz_model.predict(source=sample, imgsz=imgsz, device=device, verbose=False)
16
17 # Plot
18 plt.figure(figsize=(12, 8))
19 for i, r in enumerate(results):
20     annotated = r.plot() # BGR image with boxes + keypoints
21     plt.subplot(2, 3, i + 1)
22     plt.imshow(annotated[:, :, ::-1])
23     plt.axis('off')
24 plt.tight_layout()
25

```



- Okay Now the accuracy is quite useful.

Trying without augmentation

```

1 from ultralytics import YOLO
2
3
4 weights = Path("/content/drive/MyDrive/cow_pose_training/experiments/with_kpt_sigmas/weights/best.pt")
5 # Use a smaller model given the small dataset
6 model = YOLO(weights) # Small model - better for 251 samples
7

```

```

8 results = model.train(
9   data="/content/cow_yolo_dataset/dataset_fixed.yaml",
10  epochs=200,
11  imgsz=640,
12  batch=-1, # Auto batch
13  patience=30,
14  device=0,
15  project="/content/drive/MyDrive/cow_pose_training/experiments",
16  name="with_kpt_sigmas_no_hyps",
17 )

```

Ultralytics 8.4.12 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)  
**engine/trainer:** agnostic\_nms=False, amp=True, angle=1.0, augment=False, auto\_augment=randaugment, batch=-1, bgr

	from	n	params	module	arguments
0		-1	1	928 ultralytics.nn.modules.conv.Conv	[3, 32, 3, 2]
1		-1	1	18560 ultralytics.nn.modules.conv.Conv	[32, 64, 3, 2]
2		-1	1	29056 ultralytics.nn.modules.block.C2f	[64, 64, 1, True]
3		-1	1	73984 ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
4		-1	2	197632 ultralytics.nn.modules.block.C2f	[128, 128, 2, True]
5		-1	1	295424 ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
6		-1	2	788480 ultralytics.nn.modules.block.C2f	[256, 256, 2, True]
7		-1	1	1180672 ultralytics.nn.modules.conv.Conv	[256, 512, 3, 2]
8		-1	1	1838080 ultralytics.nn.modules.block.C2f	[512, 512, 1, True]
9		-1	1	656896 ultralytics.nn.modules.block.SPPF	[512, 512, 5]
10		-1	1	0 torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11		[-1, 6]	1	0 ultralytics.nn.modules.conv.Concat	[1]
12		-1	1	591360 ultralytics.nn.modules.block.C2f	[768, 256, 1]
13		-1	1	0 torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14		[-1, 4]	1	0 ultralytics.nn.modules.conv.Concat	[1]
15		-1	1	148224 ultralytics.nn.modules.block.C2f	[384, 128, 1]
16		-1	1	147712 ultralytics.nn.modules.conv.Conv	[128, 128, 3, 2]
17		[-1, 12]	1	0 ultralytics.nn.modules.conv.Concat	[1]
18		-1	1	493056 ultralytics.nn.modules.block.C2f	[384, 256, 1]
19		-1	1	590336 ultralytics.nn.modules.conv.Conv	[256, 256, 3, 2]
20		[-1, 9]	1	0 ultralytics.nn.modules.conv.Concat	[1]
21		-1	1	1969152 ultralytics.nn.modules.block.C2f	[768, 512, 1]
22		[15, 18, 21]	1	2929117 ultralytics.nn.modules.head.Pose	[1, [26, 3], 16, None, [128,

YOLOv8s-pose summary: 145 layers, 11,948,669 parameters, 11,948,653 gradients, 31.7 GFLOPs

Transferred 397/397 items from pretrained weights  
Freezing layer 'model.22.dfl.conv.weight'  
**AMP:** running Automatic Mixed Precision (AMP) checks...  
**AMP:** checks passed ✓  
**train:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 71.3±36.1 MB/s, size: 217.6 KB)  
**train:** Scanning /content/cow\_yolo\_dataset/labels/train.cache... 226 images, 0 backgrounds, 0 corrupt: 100% —  
**albumentations:** Blur(p=0.01, blur\_limit=(3, 7)), MedianBlur(p=0.01, blur\_limit=(3, 7)), ToGray(p=0.01, method='')  
**AutoBatch:** Computing optimal batch size for imgsz=640 at 60.0% CUDA memory utilization.  
**AutoBatch:** CUDA:0 (Tesla T4) 14.74G total, 0.12G reserved, 0.12G allocated, 14.50G free

Params	GFLOPs	GPU_mem (GB)	forward (ms)	backward (ms)	input	out
11948669	31.69	0.885	66.09	nan	(1, 3, 640, 640)	l:
11948669	63.38	1.724	26.1	nan	(2, 3, 640, 640)	l:
11948669	126.8	2.900	33.81	nan	(4, 3, 640, 640)	l:
11948669	253.5	5.184	62.88	nan	(8, 3, 640, 640)	l:
11948669	507.1	9.278	110.8	nan	(16, 3, 640, 640)	l:

**AutoBatch:** Using batch-size 15 for CUDA:0 9.09G/14.74G (62%) ✓  
**train:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 71.5±27.2 MB/s, size: 181.0 KB)  
**train:** Scanning /content/cow\_yolo\_dataset/labels/train.cache... 226 images, 0 backgrounds, 0 corrupt: 100% —  
**albumentations:** Blur(p=0.01, blur\_limit=(3, 7)), MedianBlur(p=0.01, blur\_limit=(3, 7)), ToGray(p=0.01, method='')  
**val:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 38.2±11.2 MB/s, size: 198.6 KB)  
**val:** Scanning /content/cow\_yolo\_dataset/labels/val.cache... 25 images, 0 backgrounds, 0 corrupt: 100% —  
Plotting labels to /content/drive/MyDrive/cow\_pose\_training/experiments/with\_kpt\_sigmas\_no\_hyps2/labels.jpg...  
**optimizer:** 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', ''  
**optimizer:** AdamW(lr=0.002, momentum=0.9) with parameter groups 63 weight(decay=0.0), 73 weight(decay=0.00046875)  
Image sizes 640 train, 640 val  
Using 2 dataloader workers  
Logging results to /content/drive/MyDrive/cow\_pose\_training/experiments/with\_kpt\_sigmas\_no\_hyps2  
Starting training for 200 epochs...

```

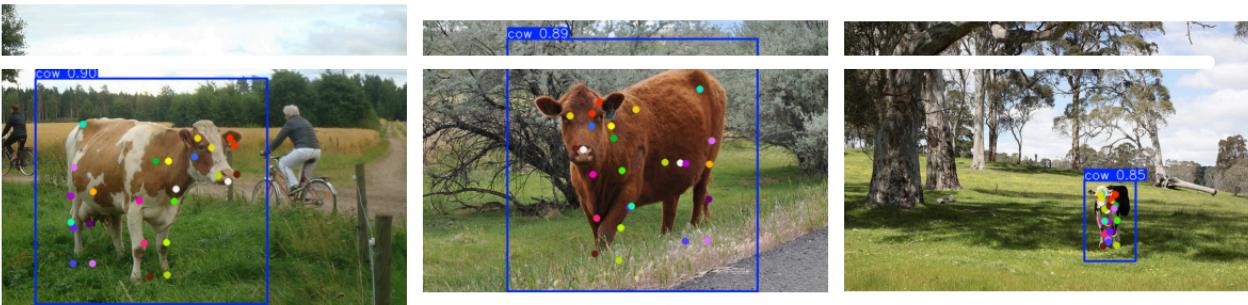
1
2 import random
3 from pathlib import Path
4 import matplotlib.pyplot as plt
5 from ultralytics import YOLO
6
7 weights = Path("/content/drive/MyDrive/cow_pose_training/experiments/with_kpt_sigmas_no_hyps2/weights/best.p
8 # Load model
9 viz_model = YOLO(str(weights))
10
11 # Pick a few validation images
12 val_dir = Path(DATASET_DIR) / 'images' / 'val'
13 val_images = list(val_dir.rglob('*.*')) + list(val_dir.rglob('*.*'))
14 sample = random.sample(val_images, k=min(6, len(val_images)))
15
16 results = viz_model.predict(source=sample, imgsz=imgsz, device=device, verbose=False)
17

```

```

18 # Plot
19 plt.figure(figsize=(12, 8))
20 for i, r in enumerate(results):
21     annotated = r.plot() # BGR image with boxes + keypoints
22     plt.subplot(2, 3, i + 1)
23     plt.imshow(annotated[:, :, ::-1])
24     plt.axis('off')
25 plt.tight_layout()
26

```



## Experiment:

Model Arch: Pose - Medium

Weights: COCO Hyper-Parameters: Default

```

1 from ultralytics import YOLO
2
3 # Use a smaller model given the small dataset
4 model = YOLO("yolov8m-pose.pt") # Small model - better for 251 samples
5
6 results = model.train(
7     data="/content/cow_yolo_dataset/dataset_fixed.yaml",
8     epochs=200,
9     imgsz=640,
10    batch=32, # Auto batch
11    patience=30,
12    device=0,
13    project="/content/drive/MyDrive/cow_pose_training/experiments",
14    name="with_kpt_sigmas_no_hyps_medium_size",
15 )

```

Ultralytics 8.4.12 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)

**engine/trainer:** agnostic\_nms=False, amp=True, angle=1.0, augment=False, auto\_augment=randaugment, batch=32, bgr=Overriding model.yaml kpt\_shape=[17, 3] with kpt\_shape=[26, 3]

	n	params	module	arguments
0	-1	1392	ultralytics.nn.modules.conv.Conv	[3, 48, 3, 2]
1	-1	41664	ultralytics.nn.modules.conv.Conv	[48, 96, 3, 2]
2	-1	111360	ultralytics.nn.modules.block.C2f	[96, 96, 2, True]
3	-1	166272	ultralytics.nn.modules.conv.Conv	[96, 192, 3, 2]
4	-1	813312	ultralytics.nn.modules.block.C2f	[192, 192, 4, True]
5	-1	664320	ultralytics.nn.modules.conv.Conv	[192, 384, 3, 2]
6	-1	3248640	ultralytics.nn.modules.block.C2f	[384, 384, 4, True]
7	-1	1991808	ultralytics.nn.modules.conv.Conv	[384, 576, 3, 2]
8	-1	3985920	ultralytics.nn.modules.block.C2f	[576, 576, 2, True]
9	-1	831168	ultralytics.nn.modules.block.SPPF	[576, 576, 5]

```

10      -1 1      0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
11      [-1, 6] 1      0 ultralytics.nn.modules.conv.Concat [1]
12      -1 2 1993728 ultralytics.nn.modules.block.C2f [960, 384, 2]
13      -1 1      0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
14      [-1, 4] 1      0 ultralytics.nn.modules.conv.Concat [1]
15      -1 2 517632 ultralytics.nn.modules.block.C2f [576, 192, 2]
16      -1 1 332160 ultralytics.nn.modules.conv.Conv [192, 192, 3, 2]
17      [-1, 12] 1      0 ultralytics.nn.modules.conv.Concat [1]
18      -1 2 1846272 ultralytics.nn.modules.block.C2f [576, 384, 2]
19      -1 1 1327872 ultralytics.nn.modules.conv.Conv [384, 384, 3, 2]
20      [-1, 9] 1      0 ultralytics.nn.modules.conv.Concat [1]
21      -1 2 4207104 ultralytics.nn.modules.block.C2f [960, 576, 2]
22      [15, 18, 21] 1 4768669 ultralytics.nn.modules.head.Pose [1, [26, 3], 16, None, [192,
YOLOv8m-pose summary: 185 layers, 26,849,293 parameters, 26,849,277 gradients, 83.0 GFLOPs

```

Transferred 481/517 items from pretrained weights

Freezing layer 'model.22.dfl.conv.weight'

**AMP:** running Automatic Mixed Precision (AMP) checks...

**AMP:** checks passed ✓

**train:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 2105.7±636.7 MB/s, size: 184.5 KB)

**train:** Scanning /content/cow\_yolo\_dataset/labels/train.cache... 226 images, 0 backgrounds, 0 corrupt: 100% —

**albumentations:** Blur(p=0.01, blur\_limit=(3, 7)), MedianBlur(p=0.01, blur\_limit=(3, 7)), ToGray(p=0.01, method='

**val:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 979.7±1004.6 MB/s, size: 230.9 KB)

**val:** Scanning /content/cow\_yolo\_dataset/labels/val.cache... 25 images, 0 backgrounds, 0 corrupt: 100% —

Plotting labels to /content/drive/MyDrive/cow\_pose\_training/experiments/with\_kpt\_sigmas\_no\_hyps\_medium\_size/labels

**optimizer:** 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', ''

**optimizer:** AdamW(lr=0.002, momentum=0.9) with parameter groups 83 weight(decay=0.0), 93 weight(decay=0.0005), 9;

Image sizes 640 train, 640 val

Using 2 dataloader workers

Logging results to /content/drive/MyDrive/cow\_pose\_training/experiments/with\_kpt\_sigmas\_no\_hyps\_medium\_size

Starting training for 200 epochs...

Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size
1/200	13.5G	2.484	10.38	0.7135	4.434	2.87	5	640: 100% —
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	Pose(P
	all	25	25	0.0936	0.52	0.075	0.0274	R
								0
Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size
2/200	13.8G	1.579	10.83	0.6828	1.67	1.955	2	640: 100% —
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	Pose(P
	all	25	25	0.768	0.8	0.826	0.436	R
								0

```

1 weights = Path("/content/drive/MyDrive/cow_pose_training/experiments/with_kpt_sigmas_no_hyps_medium_size/wei
2 viz_model = YOLO(str(weights))
3
4 # Pick a few validation images
5 val_dir = Path(DATASET_DIR) / 'images' / 'val'
6 val_images = list(val_dir.glob('*.*')) + list(val_dir.glob('*.*'))
7 sample = random.sample(val_images, k=min(6, len(val_images)))
8
9 results = viz_model.predict(source=sample, imgsz=imgsz, device=device, verbose=False)
10
11 # Plot
12 plt.figure(figsize=(12, 8))
13 for i, r in enumerate(results):
14     annotated = r.plot() # BGR image with boxes + keypoints
15     plt.subplot(2, 3, i + 1)
16     plt.imshow(annotated[:, :, ::-1])
17     plt.axis('off')
18 plt.tight_layout()
19

```



## Experiment:

Model Arch: Pose - Medium

Weights: COCO Hyper-Parameters: Tuned

```
degrees=10.0,
translate=0.1,
scale=0.3, # Reduced
fliplr=0.5,
mosaic=0.5,
mixup=0.0, # Disable

# Regularization
dropout=0.1,

# Learning rate
lr0=0.01,
lrf=0.01,
```

```
1 from ultralytics import YOLO
2
3 # Use a smaller model given the small dataset
4 model = YOLO("yolov8m-pose.pt") # Small model - better for 251 samples
5
6 results = model.train(
7     data="/content/cow_yolo_dataset/dataset_fixed.yaml",
8     epochs=500,
9     imgsz=640,
10    batch=-1, # Auto batch
11    patience=30,
12    device=0,
13    project="/content/drive/MyDrive/cow_pose_training/experiments",
14    name="with_kpt_sigmas_medium",
15
16    # Hyper-parameters
```

```

17    degrees=10.0,
18    translate=0.1,
19    scale=0.3, # Reduced
20    fliplr=0.5,
21    mosaic=0.5,
22    mixup=0.0, # Disable
23
24    # Regularization
25    dropout=0.1,
26
27    # Learning rate
28    lr0=0.01,
29    lrf=0.01,
30 )

```

Ultralytics 8.4.12 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)

**engine/trainer:** agnostic\_nms=False, amp=True, angle=1.0, augment=False, auto\_augment=randaugment, batch=-1, bgr=Overriding model.yaml kpt\_shape=[17, 3] with kpt\_shape=[26, 3]

	from	n	params	module	arguments
0	-1	1	1392	ultralytics.nn.modules.conv.Conv	[3, 48, 3, 2]
1	-1	1	41664	ultralytics.nn.modules.conv.Conv	[48, 96, 3, 2]
2	-1	2	111360	ultralytics.nn.modules.block.C2f	[96, 96, 2, True]
3	-1	1	166272	ultralytics.nn.modules.conv.Conv	[96, 192, 3, 2]
4	-1	4	813312	ultralytics.nn.modules.block.C2f	[192, 192, 4, True]
5	-1	1	664320	ultralytics.nn.modules.conv.Conv	[192, 384, 3, 2]
6	-1	4	3248640	ultralytics.nn.modules.block.C2f	[384, 384, 4, True]
7	-1	1	1991808	ultralytics.nn.modules.conv.Conv	[384, 576, 3, 2]
8	-1	2	3985920	ultralytics.nn.modules.block.C2f	[576, 576, 2, True]
9	-1	1	831168	ultralytics.nn.modules.block.SPPF	[576, 576, 5]
10	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
12	-1	2	1993728	ultralytics.nn.modules.block.C2f	[960, 384, 2]
13	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
15	-1	2	517632	ultralytics.nn.modules.block.C2f	[576, 192, 2]
16	-1	1	332160	ultralytics.nn.modules.conv.Conv	[192, 192, 3, 2]
17	[-1, 12]	1	0	ultralytics.nn.modules.conv.Concat	[1]
18	-1	2	1846272	ultralytics.nn.modules.block.C2f	[576, 384, 2]
19	-1	1	1327872	ultralytics.nn.modules.conv.Conv	[384, 384, 3, 2]
20	[-1, 9]	1	0	ultralytics.nn.modules.conv.Concat	[1]
21	-1	2	4207104	ultralytics.nn.modules.block.C2f	[960, 576, 2]
22	[15, 18, 21]	1	4768669	ultralytics.nn.modules.head.Pose	[1, [26, 3], 16, None, [192,

YOLOv8m-pose summary: 185 layers, 26,849,293 parameters, 26,849,277 gradients, 83.0 GFLOPs

Transferred 481/517 items from pretrained weights

Freezing layer 'model.22.dfl.conv.weight'

**AMP:** running Automatic Mixed Precision (AMP) checks...

**AMP:** checks passed ✅

**train:** Fast image access ✅ (ping: 0.0±0.0 ms, read: 1150.4±1095.8 MB/s, size: 184.5 KB)

**train:** Scanning /content/cow\_yolo\_dataset/labels/train.cache... 226 images, 0 backgrounds, 0 corrupt: 100% —

**albumentations:** Blur(p=0.01, blur\_limit=(3, 7)), MedianBlur(p=0.01, blur\_limit=(3, 7)), ToGray(p=0.01, method='

**AutoBatch:** Computing optimal batch size for imgsz=640 at 60.0% CUDA memory utilization.

**AutoBatch:** CUDA:0 (Tesla T4) 14.74G total, 1.28G reserved, 0.99G allocated, 12.47G free

Params	GFLOPs	GPU_mem (GB)	forward (ms)	backward (ms)	input	out
26849293	83.01	2.307	29.93	nan	(1, 3, 640, 640)	l:
26849293	166	3.735	39.16	nan	(2, 3, 640, 640)	l:
26849293	332	5.520	64.94	nan	(4, 3, 640, 640)	l:
26849293	664.1	9.469	94.29	nan	(8, 3, 640, 640)	l:
26849293	1328	16.498	170.2	nan	(16, 3, 640, 640)	l:

**AutoBatch:** Using batch-size 5 for CUDA:0 8.77G/14.74G (60%) ✅

**train:** Fast image access ✅ (ping: 0.0±0.0 ms, read: 591.8±1074.6 MB/s, size: 233.9 KB)

**train:** Scanning /content/cow\_yolo\_dataset/labels/train.cache... 226 images, 0 backgrounds, 0 corrupt: 100% —

**albumentations:** Blur(p=0.01, blur\_limit=(3, 7)), MedianBlur(p=0.01, blur\_limit=(3, 7)), ToGray(p=0.01, method='

**val:** Fast image access ✅ (ping: 0.0±0.0 ms, read: 482.0±63.5 MB/s, size: 202.7 KB)

**val:** Scanning /content/cow\_yolo\_dataset/labels/val.cache... 25 images, 0 backgrounds, 0 corrupt: 100% —

Plotting labels to /content/drive/MyDrive/cow\_pose\_training/experiments/with\_kpt\_sigmas\_medium/labels.jpg...

**optimizer:** 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', ..

**optimizer:** AdamW(lr=0.002, momentum=0.9) with parameter groups 83 weight(decay=0.0), 93 weight(decay=0.00050781):

Image sizes 640 train, 640 val

Using 2 dataloader workers

Logging results to /content/drive/MyDrive/cow\_pose\_training/experiments/with\_kpt\_sigmas\_medium

Starting training for 500 epochs

```

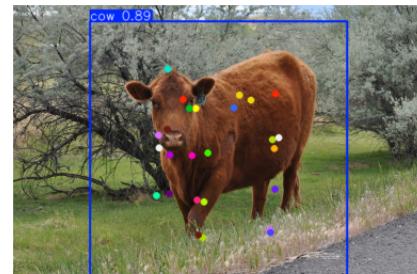
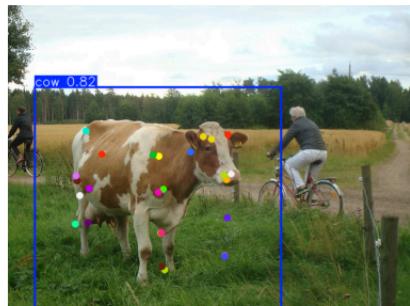
1 weights = Path("/content/drive/MyDrive/cow_pose_training/experiments/with_kpt_sigmas_medium/weights/best.pt")
2 viz_model = YOLO(str(weights))
3
4 # Pick a few validation images
5 val_dir = Path(DATASET_DIR) / 'images' / 'val'
6 val_images = list(val_dir.rglob('*.*')) + list(val_dir.rglob('*.*'))
7 sample = random.sample(val_images, k=min(6, len(val_images)))
8
9 results = viz_model.predict(source=sample, imgsz=imgsz, device=device, verbose=False)
10
11 # Plot
12 plt.figure(figsize=(12, 8))
13 for i, r in enumerate(results):

```

```

14 annotated = r.plot() # BGR image with boxes + keypoints
15 plt.subplot(2, 3, i + 1)
16 plt.imshow(annotated[:, :, ::-1])
17 plt.axis('off')
18 plt.tight_layout()
19

```



## ▼ Yolo26

Model Arch: Pose - Medium

Weights: COCO Hyper-Parameters: -

```

1 from ultralytics import YOLO
2
3 # Use a smaller model given the small dataset
4 model = YOLO("yolo26n-pose.pt") # load a pretrained model (recommended for training)
5
6 results = model.train(
7     data="/content/cow_yolo_dataset/dataset_fixed.yaml",
8     epochs=500,
9     imgsz=640,
10    batch=-1, # Auto batch
11    patience=30,
12    device=0,
13    project="/content/drive/MyDrive/cow_pose_training/experiments",
14    name="with_kpt_sigmas_yolo26m",
15
16    # Hyper-parameters
17    degrees=10.0,
18    translate=0.1,
19    scale=0.3, # Reduced
20    fliplr=0.5,
21    mosaic=0.5,
22    mixup=0.0, # Disable
23
24    # Regularization
25    dropout=0.1,
26
27    # Learning rate
28    lr0=0.01,
29    lrf=0.01,
30 )

```

```
Downloading https://github.com/ultralytics/assets/releases/download/v8.4.0/yolo26n-pose.pt to 'yolo26n-pose.pt'
Ultralytics 8.4.12 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
engine/trainer: agnostic_nms=False, amp=True, angle=1.0, augment=False, auto_augment=randaugment, batch=-1, bgr=
Overriding model.yaml kpt_shape=[17, 3] with kpt_shape=[26, 3]

      from    n   params  module                                arguments
0           -1 1       464 ultralytics.nn.modules.conv.Conv     [3, 16, 3, 2]
1           -1 1      4672 ultralytics.nn.modules.conv.Conv     [16, 32, 3, 2]
2           -1 1      6640 ultralytics.nn.modules.block.C3k2     [32, 64, 1, False, 0.25]
3           -1 1      36992 ultralytics.nn.modules.conv.Conv     [64, 64, 3, 2]
4           -1 1      26080 ultralytics.nn.modules.block.C3k2     [64, 128, 1, False, 0.25]
5           -1 1     147712 ultralytics.nn.modules.conv.Conv     [128, 128, 3, 2]
6           -1 1      87040 ultralytics.nn.modules.block.C3k2     [128, 128, 1, True]
7           -1 1      295424 ultralytics.nn.modules.conv.Conv     [128, 256, 3, 2]
8           -1 1      346112 ultralytics.nn.modules.block.C3k2     [256, 256, 1, True]
9           -1 1      164608 ultralytics.nn.modules.block.SPPF     [256, 256, 5, 3, True]
10          -1 1     249728 ultralytics.nn.modules.block.C2PSA     [256, 256, 1]
11          -1 1       0 torch.nn.modules.upsampling.Upsample     [None, 2, 'nearest']
12         [-1, 6] 1       0 ultralytics.nn.modules.conv.Concat     [1]
13          -1 1     119808 ultralytics.nn.modules.block.C3k2     [384, 128, 1, True]
14          -1 1       0 torch.nn.modules.upsampling.Upsample     [None, 2, 'nearest']
15         [-1, 4] 1       0 ultralytics.nn.modules.conv.Concat     [1]
16          -1 1      34304 ultralytics.nn.modules.block.C3k2     [256, 64, 1, True]
17          -1 1      36992 ultralytics.nn.modules.conv.Conv     [64, 64, 3, 2]
18         [-1, 13] 1       0 ultralytics.nn.modules.conv.Concat     [1]
19          -1 1      95232 ultralytics.nn.modules.block.C3k2     [192, 128, 1, True]
20          -1 1     147712 ultralytics.nn.modules.conv.Conv     [128, 128, 3, 2]
21        [-1, 10] 1       0 ultralytics.nn.modules.conv.Concat     [1]
22          -1 1      463104 ultralytics.nn.modules.block.C3k2     [384, 256, 1, True, 0.5, True]
23      [16, 19, 22] 1     2361570 ultralytics.nn.modules.head.Pose26     [1, [26, 3], 1, True, [64, 1]

YOLO26n-pose summary: 363 layers, 4,624,194 parameters, 4,624,194 gradients, 14.6 GFLOPs
```

Transferred 795/879 items from pretrained weights

**AMP:** running Automatic Mixed Precision (AMP) checks...

**AMP:** checks passed ✓

**train:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 2591.3±1136.8 MB/s, size: 184.5 KB)

**train:** Scanning /content/cow\_yolo\_dataset/labels/train.cache... 226 images, 0 backgrounds, 0 corrupt: 100% —

**albumentations:** Blur(p=0.01, blur\_limit=(3, 7)), MedianBlur(p=0.01, blur\_limit=(3, 7)), ToGray(p=0.01, method='

**AutoBatch:** Computing optimal batch size for imgsz=640 at 60.0% CUDA memory utilization.

**AutoBatch:** CUDA:0 (Tesla T4) 14.74G total, 1.97G reserved, 1.00G allocated, 11.77G free

Params	GFLOPs	GPU_mem (GB)	forward (ms)	backward (ms)	input	out
4624194	14.61	4.240	64.63	nan	(1, 3, 640, 640)	l:
4624194	29.23	4.440	58.35	nan	(2, 3, 640, 640)	l:
4624194	58.45	4.536	50.9	nan	(4, 3, 640, 640)	l:
4624194	116.9	6.197	61.7	nan	(8, 3, 640, 640)	l:
4624194	233.8	9.855	106.9	nan	(16, 3, 640, 640)	l:

**AutoBatch:** Using batch-size 9 for CUDA:0 9.90G/14.74G (67%) ✓

**train:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 2908.1±687.4 MB/s, size: 233.9 KB)

**train:** Scanning /content/cow\_yolo\_dataset/labels/train.cache... 226 images, 0 backgrounds, 0 corrupt: 100% —

**albumentations:** Blur(p=0.01, blur\_limit=(3, 7)), MedianBlur(p=0.01, blur\_limit=(3, 7)), ToGray(p=0.01, method='

**val:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 682.5±493.3 MB/s, size: 202.7 KB)

**val:** Scanning /content/cow\_yolo\_dataset/labels/val.cache... 25 images, 0 backgrounds, 0 corrupt: 100% —

Plotting labels to /content/drive/MyDrive/cow\_pose\_training/experiments/with\_kpt\_sigmas\_yolo26m/labels.jpg...

**optimizer:** 'optimizer=auto' found, ignoring 'lr=0.01' and 'momentum=0.937' and determining best 'optimizer', ..

**optimizer:** AdamW(lr=0.002, momentum=0.9) with parameter groups 126 weight(decay=0.0), 186 weight(decay=0.000492):

Image sizes 640 train, 640 val

Using 2 dataloader workers

```
1 !mv /content/drive/MyDrive/cow_pose_training/experiments/with_kpt_sigmas_yolo26m /content/drive/MyDrive/cow_
```

```
1 weights = Path("/content/drive/MyDrive/cow_pose_training/experiments/with_kpt_sigmas_yolo26n/weights/best.pt")
2 viz_model = YOLO(str(weights))
3
4 # Pick a few validation images
5 val_dir = Path(DATASET_DIR) / 'images' / 'val'
6 val_images = list(val_dir.rglob('*.*')) + list(val_dir.rglob('*.*'))
7 sample = random.sample(val_images, k=min(6, len(val_images)))
8
9 results = viz_model.predict(source=sample, imgsz=imgsz, device=device, verbose=False)
10
11 # Plot
12 plt.figure(figsize=(12, 8))
13 for i, r in enumerate(results):
14     annotated = r.plot() # BGR image with boxes + keypoints
15     plt.subplot(2, 3, i + 1)
16     plt.imshow(annotated[:, :, ::-1])
17     plt.axis('off')
18 plt.tight_layout()
19
```



```

1 from ultralytics import YOLO
2
3 # Use a smaller model given the small dataset
4 model = YOLO("yolo26m-pose.pt") # load a pretrained model (recommended for training)
5
6 results = model.train(
7     data="/content/cow_yolo_dataset/dataset_fixed.yaml",
8     epochs=500,
9     imgsz=640,
10    batch=-1, # Auto batch
11    patience=30,
12    device=0,
13    project="/content/drive/MyDrive/cow_pose_training/experiments",
14    name="with_kpt_sigmas_yolo26m",
15
16    # Hyper-parameters
17    degrees=10.0,
18    translate=0.1,
19    scale=0.3, # Reduced
20    fliplr=0.5,
21    mosaic=0.5,
22    mixup=0.0, # Disable
23
24    # Regularization
25    dropout=0.1,
26
27    # Learning rate
28    lr0=0.01,
29    lrf=0.01,
30 )

```

Downloading <https://github.com/ultralytics/assets/releases/download/v8.4.0/yolo26m-pose.pt> to 'yolo26m-pose.pt'  
Ultralytics 8.4.12 🐾 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)  
**engine/trainer:** agnostic\_nms=False, amp=True, angle=1.0, augment=False, auto\_augment=randaugment, batch=-1, bgr\_overrides=[{"name": "yolo26m", "r": 128, "g": 128, "b": 128}], device=0, engine='pyt', fp16=False, half=False, max\_epochs=500, model='yolo26m-pose.pt', model.yaml=kpt\_shape=[17, 3], model\_type='pose', nms=True, num\_classes=1, project='/content/drive/MyDrive/cow\_pose\_training/experiments', save=True, save\_period=1, save\_suffix='', seed=42, start\_epoch=1, train=True, weights='yolo26m-pose.pt', workers=4  
Overriding model.yaml kpt\_shape=[17, 3] with kpt\_shape=[26, 3]

	from	n	params	module	arguments
0	-1	1	1856	ultralytics.nn.modules.conv.Conv	[3, 64, 3, 2]
1	-1	1	73984	ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
2	-1	1	111872	ultralytics.nn.modules.block.C3k2	[128, 256, 1, True, 0.25]
3	-1	1	590336	ultralytics.nn.modules.conv.Conv	[256, 256, 3, 2]
4	-1	1	444928	ultralytics.nn.modules.block.C3k2	[256, 512, 1, True, 0.25]
5	-1	1	2360320	ultralytics.nn.modules.conv.Conv	[512, 512, 3, 2]
6	-1	1	1380352	ultralytics.nn.modules.block.C3k2	[512, 512, 1, True]
7	-1	1	2360320	ultralytics.nn.modules.conv.Conv	[512, 512, 3, 2]
8	-1	1	1380352	ultralytics.nn.modules.block.C3k2	[512, 512, 1, True]

```

9          -1 1    656896 ultralytics.nn.modules.block.SPPF      [512, 512, 5, 3, True]
10         -1 1    990976 ultralytics.nn.modules.block.C2PSA    [512, 512, 1]
11          -1 1      0 torch.nn.modules.upsampling.Upsample   [None, 2, 'nearest']
12         [-1, 6] 1      0 ultralytics.nn.modules.conv.Concat  [1]
13          -1 1    1642496 ultralytics.nn.modules.block.C3k2    [1024, 512, 1, True]
14          -1 1      0 torch.nn.modules.upsampling.Upsample   [None, 2, 'nearest']
15         [-1, 4] 1      0 ultralytics.nn.modules.conv.Concat  [1]
16          -1 1    542720 ultralytics.nn.modules.block.C3k2    [1024, 256, 1, True]
17          -1 1    590336 ultralytics.nn.modules.conv.Conv     [256, 256, 3, 2]
18         [-1, 13] 1      0 ultralytics.nn.modules.conv.Concat  [1]
19          -1 1    1511424 ultralytics.nn.modules.block.C3k2    [768, 512, 1, True]
20          -1 1    2360320 ultralytics.nn.modules.conv.Conv     [512, 512, 3, 2]
21         [-1, 10] 1      0 ultralytics.nn.modules.conv.Concat  [1]
22          -1 1    1974784 ultralytics.nn.modules.block.C3k2    [1024, 512, 1, True, 0.5, Tri
23         [16, 19, 22] 1  6867042 ultralytics.nn.modules.head.Pose26 [1, [26, 3], 1, True, [256, !]

YOLO26m-pose summary: 383 layers, 25,841,314 parameters, 25,841,314 gradients, 92.7 GFLOPs

```

Transferred 855/939 items from pretrained weights

**AMP:** running Automatic Mixed Precision (AMP) checks...

**AMP:** checks passed ✓

**train:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 1590.5±716.1 MB/s, size: 184.5 KB)

**train:** Scanning /content/cow\_yolo\_dataset/labels/train.cache... 226 images, 0 backgrounds, 0 corrupt: 100% —

**albumentations:** Blur(p=0.01, blur\_limit=(3, 7)), MedianBlur(p=0.01, blur\_limit=(3, 7)), ToGray(p=0.01, method='

**AutoBatch:** Computing optimal batch size for imgsz=640 at 60.0% CUDA memory utilization.

**AutoBatch:** CUDA:0 (Tesla T4) 14.74G total, 2.06G reserved, 1.18G allocated, 11.50G free

Params	GFLOPs	GPU mem (GB)	forward (ms)	backward (ms)	input	out
25841314	92.66	4.584	67.27	nan	(1, 3, 640, 640)	l:
25841314	185.3	6.287	58.01	nan	(2, 3, 640, 640)	l:
25841314	370.7	9.343	86.71	nan	(4, 3, 640, 640)	l:
25841314	741.3	15.307	149	nan	(8, 3, 640, 640)	l:
25841314	1483	26.651	245.4	nan	(16, 3, 640, 640)	l:

**AutoBatch:** Using batch-size 2 for CUDA:0 9.45G/14.74G (64%) ✓

**train:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 2811.9±633.2 MB/s, size: 233.9 KB)

**train:** Scanning /content/cow\_yolo\_dataset/labels/train.cache... 226 images, 0 backgrounds, 0 corrupt: 100% —

**albumentations:** Blur(p=0.01, blur\_limit=(3, 7)), MedianBlur(p=0.01, blur\_limit=(3, 7)), ToGray(p=0.01, method='

**val:** Fast image access ✓ (ping: 0.0±0.0 ms, read: 459.5±201.6 MB/s, size: 202.7 KB)

**val:** Scanning /content/cow\_yolo\_dataset/labels/val.cache... 25 images, 0 backgrounds, 0 corrupt: 100% —

Plotting labels to /content/drive/MyDrive/cow\_pose\_training/experiments/with\_kpt\_sigmas\_yolo26m/labels.jpg...

**optimizer:** 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', ..

**optimizer:** AdamW(lr=0.002, momentum=0.9) with parameter groups 136 weight(decay=0.0), 196 weight(decay=0.0005),

Image sizes 640 train, 640 val

Using 2 dataloader workers

```

1 weights = Path("/content/drive/MyDrive/cow_pose_training/experiments/with_kpt_sigmas_yolo26m/weights/best.pt")
2 viz_model = YOLO(str(weights))
3
4 # Pick a few validation images
5 val_dir = Path(DATASET DIR) / 'images' / 'val'

```