Dataintelliage

Pose detection and Action recognition

Detecting the pose of human being and recognizing the action through different keypoints on the body .we can detect pose and action of animals also ：

1. What is pose estimation?

In the evolving landscape of computer vision, pose estimation stands out as a pivotal innovation, transforming how we understand and interact with visual data. Ultralytics YOLOv8 is at the forefront of this transformation, providing a powerful tool that captures the subtleties of object orientation and movement within images.

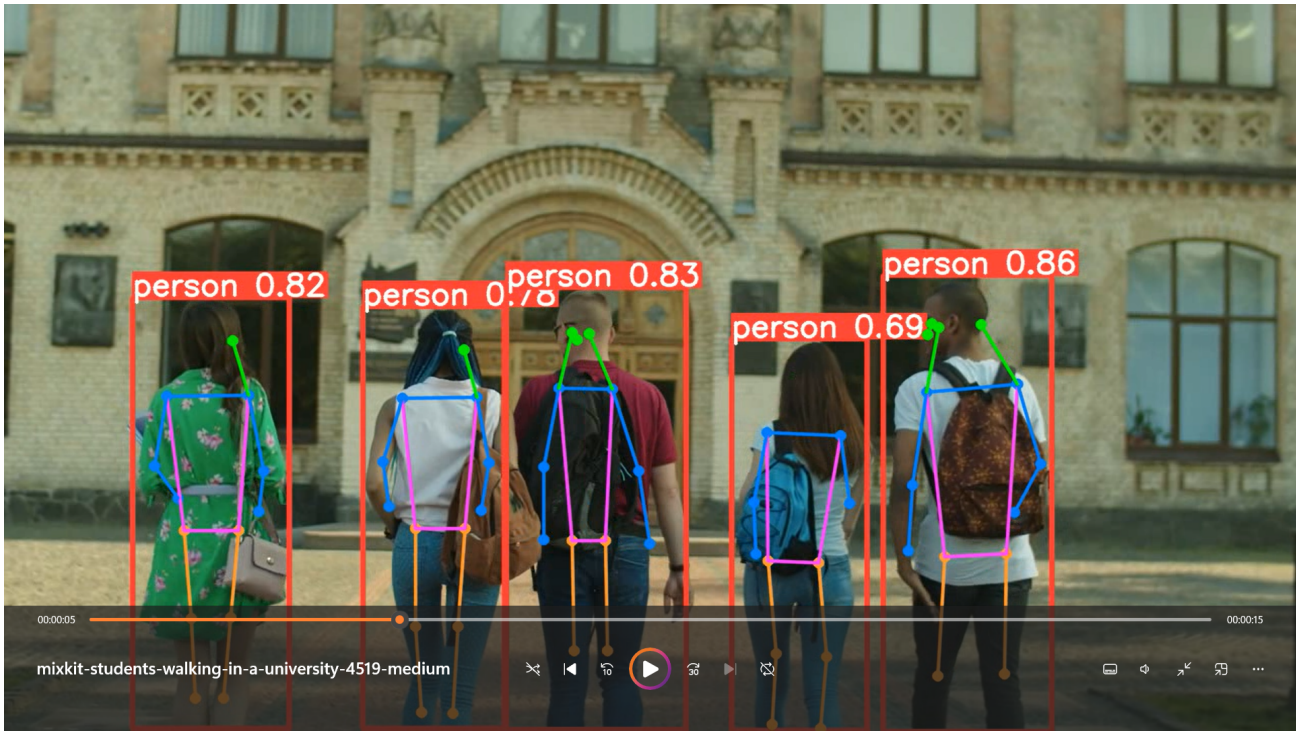
2.How it works

Pose estimation is a task that involves identifying the location of specific points in an image, usually referred to as keypoints. The keypoints can represent various parts of the object such as joints, landmarks, or other distinctive features. The locations of the keypoints are usually represented as a set of 2D [x, y] or 3D [x, y, visible] coordinates.

The output of a pose estimation model is a set of points that represent the keypoints on an object in the image, usually along with the confidence scores for each point. Pose estimation is a good choice when you need to identify specific parts of an object in a scene, and their location in relation to each other.

4. Pose estimation

First you have to make a new environment and download all the dependencies such as ultralytics which contains all the requirement if you want to use your gpu you have to enable your gpu, to enable your gpu first you have to install cuda enable though cuDnn and then install pytorch gpu enable. Here we make pose estimation with yolov8 there different model like volov8n(nano), yolov8l(large), etc. Here we are using YOLOv8n for pose estimation.



5. Pose estimation with custom data

we can also make our own model like yolov8n by using our own data though yolo trainer with your different photos like is original model there is no class of running here we train our model to recognize running ,walking and person talking on phone

First make evironment

Then install all depencencies

In [1]:
```
import ultralytics
```

In [2]:
```
from ultralytics import YOLO
```

#Download pre-trained model for pose detection

In [3]:
```
model = YOLO('yolov8n-pose.pt')
```

```
Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n-pose.pt (https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n-pose.pt) to 'yolov8n-pose.pt'...

100%|████████████████████████████████████████████████████████████████████████| 6.51M/6.51M [00:06<00:00, 1.10MB/s]
```

On photo

In [5]:
```
source = "https://yesofcorsa.com/wp-content/uploads/2018/03/Traffic-Jam-Wallpa
model.predict(source, save=True,imgsz=320, conf=0.5)
```

```
Downloading https://yesofcorsa.com/wp-content/uploads/2018/03/Traffic-Jam-Wallpaper-HD.jpg (https://yesofcorsa.com/wp-content/uploads/2018/03/Traffic-Jam-Wallpaper-HD.jpg) to 'Traffic-Jam-Wallpaper-HD.jpg'...

100%|████████████████████████████████████████████████████████████████████████| 3.58M/3.58M [00:01<00:00, 2.16MB/s]

image 1/1 C:\Users\Saurav Kumar\Traffic-Jam-Wallpaper-HD.jpg: 224x320 (no detections), 142.2ms
Speed: 0.0ms preprocess, 142.2ms inference, 30.0ms postprocess per image at shape (1, 3, 224, 320)
Results saved to runs\pose\predict
```

Out[5]:
```
[ultralytics.engine.results.Results object with attributes:

 boxes: ultralytics.engine.results.Boxes object
 keypoints: ultralytics.engine.results.Keypoints object
 masks: None
 names: {0: 'person'}
```

On video

In [6]:
```
source = "D:\mixkit-students-walking-in-a-university-4519-medium.mp4"
model.predict(source, save=True,imgsz=320, conf=0.5)
```

```
WARNING ⚠️ inference results will accumulate in RAM unless `stream=True` i
s passed, causing potential out-of-memory
errors for large sources or long-running streams and videos. See https://d
ocs.ultralytics.com/modes/predict/ (https://docs.ultralytics.com/modes/pre
dict/) for help.

Example:
    results = model(source=..., stream=True)  # generator of Results objec
ts
    for r in results:
        boxes = r.boxes  # Boxes object for bbox outputs
        masks = r.masks  # Masks object for segment masks outputs
        probs = r.probs  # Class probabilities for classification outputs

video 1/1 (1/603) D:\mixkit-students-walking-in-a-university-4519-medium.m
p4: 192x320 5 persons, 110.2ms
video 1/1 (2/603) D:\mixkit-students-walking-in-a-university-4519-medium.m
```

On web-cam

In [ ]:
```python
import cv2
from ultralytics import YOLO
model = YOLO('yolov8n-pose.pt')

video_path = 0
cap = cv2.VideoCapture(video_path)

while cap.isOpened():

    success, frame = cap.read()

    if success:
        result = model(frame, save=True)

        annotation_frame = result[0].plot()

        cv2.imshow("YOLOv8 inference", annotation_frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    else:
        break

cap.release()
cv2.destroyAllWindows()
```

In [ ]:

pose detection on custom data

In [1]:
```
import ultralytics
```

In [2]:
```
from ultralytics import YOLO
```

In [4]:
```
model = YOLO('6de2cbda-ddd1-4427-b4c9-825235ec8e56.pt')
```

In [6]:
```
source = r"C:\Users\Saurav Kumar\Downloads\F1AF1AE7-506B-8DC1-EBB4077441043534_
model.predict(source, save=True,imgsz=320, conf=0.5)
```

```
image 1/1 C:\Users\Saurav Kumar\Downloads\F1AF1AE7-506B-8DC1-EBB4077441043
534_page-header-xs-1x.jpg: 320x224 1 running, 142.0ms
Speed: 15.6ms preprocess, 142.0ms inference, 157.1ms postprocess per image
at shape (1, 3, 320, 224)
Results saved to runs\pose\predict6
```

Out[6]:
```
[ultralytics.engine.results.Results object with attributes:

 boxes: ultralytics.engine.results.Boxes object
 keypoints: ultralytics.engine.results.Keypoints object
 masks: None
 names: {0: 'person', 1: 'standing', 2: 'running', 3: 'talking on phone'}
 orig_img: array([[[  0, 117,  90],
         [  1, 118,  91],
         [  4, 117,  90],
         ...,
         [106, 226, 195],
         [ 98, 221, 189],
         [ 91, 214, 182]]
```

In [14]:
```
source = "D:\mixkit-man-and-woman-jogging-through-a-park-in-the-city-40754-med
model.predict(source, save=True,imgsz=320, conf=0.5)
```

```
WARNING ⚠️ inference results will accumulate in RAM unless `stream=True` i
s passed, causing potential out-of-memory
errors for large sources or long-running streams and videos. See https://d
ocs.ultralytics.com/modes/predict/ (https://docs.ultralytics.com/modes/pre
dict/) for help.

Example:
    results = model(source=..., stream=True)  # generator of Results objec
ts
    for r in results:
        boxes = r.boxes  # Boxes object for bbox outputs
        masks = r.masks  # Masks object for segment masks outputs
        probs = r.probs  # Class probabilities for classification outputs

video 1/1 (1/304) D:\mixkit-man-and-woman-jogging-through-a-park-in-the-ci
ty-40754-medium.mp4: 192x320 (no detections), 16.0ms
video 1/1 (2/304) D:\mixkit-man-and-woman-jogging-through-a-park-in-the-ci
```

In [5]:
```
import cv2
from ultralytics import YOLO
model = YOLO('6de2cbda-ddd1-4427-b4c9-825235ec8e56.pt')

video_path = 0
cap = cv2.VideoCapture(video_path)

while cap.isOpened():

    success, frame = cap.read()

    if success:
        result = model(frame, save=True)

        annotation_frame = result[0].plot()

        cv2.imshow("YOLOv8 inference", annotation_frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    else:
        break

cap.release()
cv2.destroyAllWindows()
```

```
0: 480x640 (no detections), 157.1ms
Speed: 0.0ms preprocess, 157.1ms inference, 31.2ms postprocess per image a
t shape (1, 3, 480, 640)
Results saved to runs\pose\predict14

0: 480x640 (no detections), 15.5ms
Speed: 4.2ms preprocess, 15.5ms inference, 0.0ms postprocess per image at
shape (1, 3, 480, 640)
Results saved to runs\pose\predict14

0: 480x640 (no detections), 24.8ms
Speed: 0.0ms preprocess, 24.8ms inference, 8.5ms postprocess per image at
shape (1, 3, 480, 640)
Results saved to runs\pose\predict14

0: 480x640 (no detections), 34.6ms
Speed: 0.0ms preprocess, 34.6ms inference, 0.0ms postprocess per image at
shape (1, 3, 480, 640)
```

In [ ]: