```python
!pip install ultralytics opencv-python hmmlearn

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

# === Imports ===
import os, cv2, pickle
import numpy as np
from tqdm import tqdm
from ultralytics import YOLO
from sklearn.preprocessing import StandardScaler
from hmmlearn import hmm
from scipy.ndimage import uniform_filter1d
```

--NORMAL--

```
Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl (211.5 MB)
                                                211.5/211.5 MB 6.3 MB/s eta 0:00:00
Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.whl (56.3 MB)
                                                56.3/56.3 MB 13.3 MB/s eta 0:00:00
Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl (127.9 MB)
                                                127.9/127.9 MB 7.5 MB/s eta 0:00:00
Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl (207.5 MB)
                                                207.5/207.5 MB 5.6 MB/s eta 0:00:00
Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (21.1 MB)
                                                21.1/21.1 MB 88.0 MB/s eta 0:00:00
Downloading ultralytics_thop-2.0.14-py3-none-any.whl (26 kB)
Installing collected packages: nvidia-nvjitlink-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-cupti-cu12, nvidia-cublas-cu12, nvidia-cusparse-cu12, nvidia-cudnn-c
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
      Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
    Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-nvrtc-cu12
    Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
    Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-cupti-cu12
    Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
    Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
  Attempting uninstall: nvidia-cublas-cu12
    Found existing installation: nvidia-cublas-cu12 12.5.3.2
    Uninstalling nvidia-cublas-cu12-12.5.3.2:
      Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
  Attempting uninstall: nvidia-cusparse-cu12
    Found existing installation: nvidia-cusparse-cu12 12.5.1.3
    Uninstalling nvidia-cusparse-cu12-12.5.1.3:
      Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
  Attempting uninstall: nvidia-cudnn-cu12
    Found existing installation: nvidia-cudnn-cu12 9.3.0.75
    Uninstalling nvidia-cudnn-cu12-9.3.0.75:
      Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
  Attempting uninstall: nvidia-cusolver-cu12
    Found existing installation: nvidia-cusolver-cu12 11.6.3.83
    Uninstalling nvidia-cusolver-cu12-11.6.3.83:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
Successfully installed hmmlearn-0.3.3 nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cuda-runtime-cu12-12.4.127 nvidia-cudnn-cu12-9.1.0.70 nvidia-cufft-cu12-11.2.1.3 nvidi
Mounted at /content/drive
Creating new Ultralytics Settings v0.0.6 file ✅
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com/quickstart/#ultralytics-settings.
```

```python
# === Parameters ===
YOLO_MODEL_PATH = "/content/drive/MyDrive/Colab Notebooks/hmm_anomaly_detection/yolov8n.pt"  # Change this if needed
WALK_DIR = "/content/drive/MyDrive/Colab Notebooks/hmm_anomaly_detection/anomaly_dataset/1_walk"    # Upload your walk videos here
ENTER_DIR = "/content/drive/MyDrive/Colab Notebooks/hmm_anomaly_detection/anomaly_dataset/2_enter"  # Upload your enter videos here
WINDOW_SIZE = 50
FRAME_SKIP = 2
SMOOTH_KERNEL = 5  # Moving average window for smoothing

# === Feature Extraction with Smoothing and Sliding Window ===
def extract_features_sliding(video_path, model):
    cap = cv2.VideoCapture(video_path)
    if not cap.isOpened():
        return []

    w = cap.get(cv2.CAP_PROP_FRAME_WIDTH)
    h = cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
    feats = []
    frame_idx = 0

    while True:
        ret, frame = cap.read()
        if not ret:
            break

        if frame_idx % FRAME_SKIP == 0:
            res = model.predict(source=frame, verbose=False)[0]
            raw = res.boxes.data.cpu().numpy()
            person_mask = raw[:, 5].astype(int) == 0
            persons = raw[person_mask][:, :4]
            if persons.size > 0:
                x1, y1, x2, y2 = max(persons, key=lambda b: (b[2]-b[0])*(b[3]-b[1]))
                cx = ((x1 + x2) / 2) / w
                cy = ((y1 + y2) / 2) / h
                fw = (x2 - x1) / w
                fh = (y2 - y1) / h
            else:
                cx = cy = fw = fh = np.nan
            feats.append([cx, cy, fw, fh])

        frame_idx += 1

    cap.release()

    if len(feats) == 0:
        return []

    F = np.vstack(feats)
    valid = ~np.isnan(F[:, :4]).any(axis=1)
    if valid.any():
        first = np.argmax(valid)
        F[:first, :4] = F[first, :4]
    else:
```

```python
            F[:, 0:2] = 0.5
            F[:, 2:4] = 0.0

        for i in range(1, F.shape[0]):
            for j in range(4):
                if np.isnan(F[i, j]):
                    F[i, j] = F[i-1, j]

        # ✅ Apply smoothing
        for j in range(4):
            F[:, j] = uniform_filter1d(F[:, j], size=SMOOTH_KERNEL, mode='nearest')

        # Sliding window
        seqs = []
        for start in range(0, len(F) - WINDOW_SIZE + 1, 5):
            clip = F[start:start+WINDOW_SIZE]
            if clip.shape[0] == WINDOW_SIZE:
                seqs.append(clip)

        return seqs

    # === HMM Training ===
    def train_hmm(seqs, n_states=7, percentile=5.0):
        lengths = [s.shape[0] for s in seqs]
        X = np.vstack(seqs)
        scaler = StandardScaler().fit(X)
        Xs = scaler.transform(X)
        model = hmm.GaussianHMM(n_components=n_states, covariance_type='full', n_iter=100)
        model.fit(Xs, lengths)
        lls = [model.score(scaler.transform(s)) for s in seqs]
        thr = np.percentile(lls, percentile)
        return {'model': model, 'scaler': scaler, 'threshold': thr}

    # === Main Training Code ===
    model = YOLO(YOLO_MODEL_PATH)

    for label, path in {'walk': WALK_DIR, 'enter': ENTER_DIR}.items():
        all_seqs = []
        print(f"⏳ Extracting features for: {label}")
        for f in tqdm(sorted(os.listdir(path))):
            if f.endswith(".mp4"):
                full = os.path.join(path, f)
                seqs = extract_features_sliding(full, model)
                all_seqs.extend(seqs)
        print(f"🎯 {label}: {len(all_seqs)} clips")

        cfg = train_hmm(all_seqs)

        with open(f"hmm_{label}.pkl", 'wb') as f:
            pickle.dump(cfg, f)

        print(f"✅ Saved hmm_{label}.pkl")
```

```
⏳ Extracting features for: walk
100%|████████| 28/28 [01:00<00:00,  2.16s/it]
🎯 walk: 276 clips
WARNING:hmmlearn.base:Model is not converging.  Current: 29575.084174892705 is not greater than 29577.679021721156. Delta is -2.5948468284514092
✅ Saved hmm_walk.pkl
⏳ Extracting features for: enter
100%|████████| 18/18 [01:14<00:00,  4.16s/it]
🎯 enter: 840 clips
✅ Saved hmm_enter.pkl
```

```python
# Save the results back to google drive
!mkdir -p '/content/drive/MyDrive/Colab Notebooks/hmm_anomaly_detection/anomaly_result'
!cp hmm_walk.pkl '/content/drive/MyDrive/Colab Notebooks/hmm_anomaly_detection/anomaly_result/hmm_walk.pkl'
!cp hmm_enter.pkl '/content/drive/MyDrive/Colab Notebooks/hmm_anomaly_detection/anomaly_result/hmm_enter.pkl'
```

Variables   Terminal