1. **Identify ML training related to video datasets, for instance, image classification for image datasets.**

Video Classification
- Attributing some label with a given video
- Ex. sports

Action/Gesture Recognition
- Human actions/gestures within a given video

Event Detection
- Detecting specific events in a video
- Accidents on a road
- Unusual activity in a crowd

Scene Classification
- Categorizing videos based on the scene/environment in the video
- Indoor scenes, outdoor scenes, urban landscapes, etc

Object Detection/Tracking
- Locating objects of interest in a video
- Tracking these objects of interest

There are a large number of possible training related to video datasets.

2. **Identify video datasets commonly used to for tasks in 1.**

Found this github page for a list of all relevant datasets for different kinds of video based machine learning models

https://github.com/xiaobai1217/Awesome-Video-Datasets
Other large/famous video datasets
- UCF101: 13320 videos and 101 action classes
    - Good for action recognition
- HMDB51: 6849 videos and 51 action classes
- Kinetics: 400,000 videos 600 action classes
- Youtube-8M: Youtube video urls and 4716 vocab classes
    - General classification tasks
- Sports-1M: 1 million videos from 487 classes of sports
    - Good for video classification

3. **Identify ML models needed for above tasks, ideally find smaller models used.**

- Video Classification:
    - CNN-LSTM Model: Convolutional Neural Networks (CNNs) for spatial features and Long Short-Term Memory (LSTM) networks for temporal features (Might be too big for our use case)
    - 3D CNNs: Models like C3D (Convolutional 3D) or R(2+1)D

- Action/Gesture Recognition:
    - I3D (Inflated 3D ConvNet): action recognition in videos. You can use smaller variants of I3D for faster inference.
    - Temporal Convolutional Networks (TCNs): TCNs are lightweight and can be used for gesture recognition.

- Event Detection:
    - Two-Stream Networks: Combine two CNN streams (one for spatial and one for optical flow) and fuse their features to detect events. Smaller versions of CNNs can be used here.
    - Single Shot MultiBox Detector (SSD): For detecting events like accidents

- Object Detection/Tracking:
    - YOLO (You Only Look Once): YOLO models, YOLOv3-tiny or YOLOv4-tiny,
    - SORT (Simple Online and Realtime Tracking): For object tracking, SORT is a simple yet effective choice.
        - Needs to be used with a detector model as well

4. **Identify preprocessing operations applied on video datasets, for instance, images use decoding, RandomResizedCrop, RandomHorizontalFlip, ToTensor, and Normalize. For images, @rajveerb referred to MLPerf's example. Maybe look at research papers as well.**

Some papers that I need to read over/get to

- https://arxiv.org/pdf/1412.0767.pdf
- https://arxiv.org/pdf/1503.08909.pdf
- https://arxiv.org/pdf/1502.04681.pdf
- https://arxiv.org/pdf/1602.00763.pdf

In general what I have noticed for pre processing so far:

1. Video loading
2. Frame Resizing

3. Temporal sampling (for certain tasks not all)
4. Data augmentation like flipping, rotating, etc, can be applied
5. Normalization
6. Batching

Preprocessing on C3D:

- Split each video into five 2-second clips
- Each clip is randomly cropped to be 16x112x112 for both spatial and temporal jittering
- 50% chance to be randomly flipped

Detection, Tracking, and Counting Meets Drones in Crowds: A Benchmark:

- To increase diversity in training data, we randomly flip and crop the training images.
- Due to limited computation resources, we equally divide each frame into 2 × 2 patches,and use the divided 4 patches with the resolution of 960 ×540 for training

Action Recognition:

HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences

- Datasets used:
  - MSR Actions 3D Dataset
  - MSR Gesture 3D
  - MSR Daily Activity 3D
- Processing used:
  - frame size in all datasets is 320 × 240
  - divided into spatiotemporal cells, which are typically 4×3×3 (w * h * # of frames)
  - 300 projectors created
    - Projectors created using their own math that I honestly don't really understand

Rolling Rotations for Recognizing Human Actions from 3D Skeletal Data

- Datasets used:
  - Florence3D-Action
  - MSRAction Pairs
  - G3D-Gaming

- Code provided: http://ravitejav.weebly.com/rolling.html

- Processing used:
  1. Skeletal representation,
  2. Nominal curve computation using DTW
  3. Rolling and unwrapping
  4. Linear SVM classification