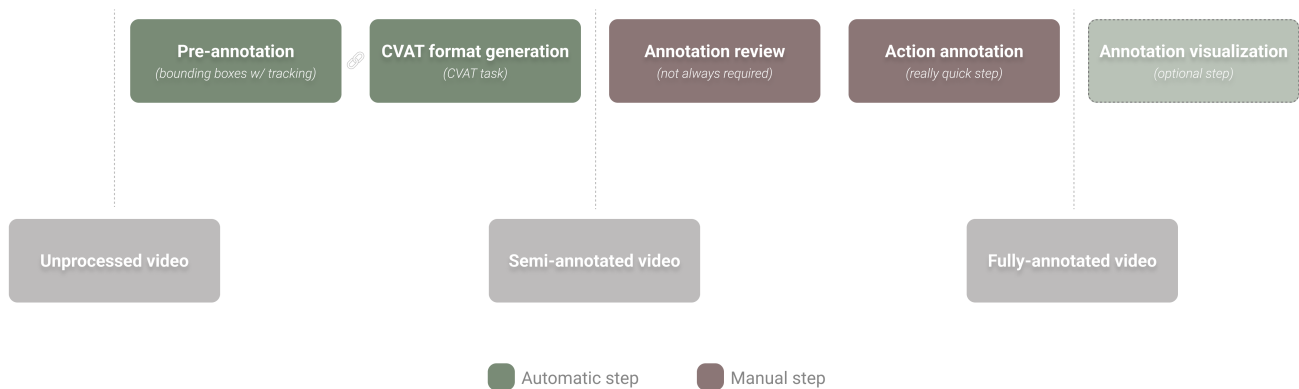# Annotation Tool v2.0 Guide

Pipeline Description

Pre-annotation: uses the chosen algorithm (*ByteTrack*, *AlphaPose* or *both*) to compute **bounding boxes for humans**, while also ensuring that they are **tracked**. Note that bounding boxes for inanimate objects (i.e., cars or fires) still need to be handcrafted, for now.

CVAT format generation: generates a *.zip* file that essentially recreates a **CVAT task** already **pre-loaded with the automatic annotations** from the previous step. This makes it super easy to upload to CVAT for further examination.

Annotation review: allows us to fix any **tracking error** (i.e., non-existent IDs, poorly attributed IDs, bounding boxes that shouldn't have been placed, etc...):

```
# NOTE: frames are 1-indexed
# NOTE: the ranges below include both ends

# NOTE: use this list to REMOVE WRONG IDs in the format
["current_object_id", start_frame, end_frame]
REMOVE_LIST = [
    ["1", 1, -1], # e.g., ID "1" will be remove from every frame
]

# NOTE: use this list to REMAP ONE ID TO ANOTHER in the format
["current_object_id", start_frame, end_frame, "correct_object_id"]
REMAP_LIST = [
    ["1", 1, -1, "2"], # e.g., ID "1" will become ID "2" in every frame
]

# sample function call
fix_tracking(remove_list = REMOVE_LIST, remap_list = REMAP_LIST)
```

<u>OR</u> to fix the **actual bounding boxes** (i.e., poorly placed or entirely non-existent bounding boxes, etc...):

1. Go to **CVAT** (**has to be through <u>https://cvat.org</u> on a supported browser** - Firefox or Chrome)
2. Click on "**Create from backup**" to upload the .zip file from the previous step (i.e., a CVAT task **pre-loaded with the automatic annotations**)
3. **Fix or add** any bounding boxes (**make sure the IDs are properly set**)
4. Once finished, **download the task** and place it in the ***/output*** folder (thus replacing the previous file)

<u>Action annotation</u>: enables us to specify the actions being performed by the people in the video:

```
'''
---------------
POSSIBLE ACTIONS
---------------
walking, running, standing, standing_up, sitting_down, jumping, riding,
climbing, lying, throwing, falling, carrying_weapon, abandoned_object,
fighting, stealing, shooting, vandalizing, fire_raising, fire_or_explosion,
road_accident
'''

# NOTE: frames are 1-indexed
# NOTE: you can use a "-1" to indicate that the action goes until the last
frame
# NOTE: the ranges below include both ends
# NOTE: there are 4 possible object classes: "fire", "object", "car" and
"human"

# NOTE: use this list to ANNOTATE ACTIONS in the format ["object_id",
start_frame, end_frame, "action"]
ACTIONS_LIST = [
    ["1", 1, -1, "shooting"], # e.g., ID "1" is "shooting" in every frame
]

# sample function call
annotate_actions(video_path = VIDEO_PATH, actions_list = ACTIONS_LIST)
```

<u>(OPTIONAL) Annotation visualization</u>: creates a video with the result of the annotation process.

Installation

1. If needed, install **ipykernel**:

```
python3 -m pip install --user ipykernel
```

2. Initialize the **virtual environment**:

```
python3 -m ipykernel install --user --name=.venv
```