

Daily Log

Thursday August 29

Drafted and sent email to the ROSE lab, requesting access to the NTU CCTV-Fights Dataset, which contains over 17 hours of surveillance camera footage with parts marked as violent.
Reviewed OpenPose and dataset generation code from my science fair project as a refresher.

Tuesday September 3

Discussed potentially using the Keras functional API with Saketh to combine his emotion detection network with my pose estimation model.
Received access to the NTU CCTV-Fights Dataset. Downloaded part of the dataset to build code around. It took a long time for even small parts of the dataset (0.3 gb zip file) to download on the FCPS Wifi, so I will try and download the rest at home.
Wrote code that accesses and can parse through a "groundtruth.json" file from the ROSE Lab website. The JSON file indexes every file in the dataset and labels the times in the video in which a fight is occurring.

Thursday September 5

Wrote code that iterates over each frame in every video in the ROSE lab dataset, creates a dictionary that contains each frame marked as either violent or non-violent based on ground truth values from the aforementioned JSON file and has other relevant information, and extensively tested the code on parts of the dataset to ensure that it runs properly and can be used for the entire dataset. Thought about potentially adding features to normalize the size of every frame and remove borders, but realized this probably won't be necessary since it shouldn't effect the statistical representation used to interpret OpenPose results.

Timeline

Date	Goal	Met
Today	Settle on a dataset to use for the project.	Yes, received access to the NTU CCTV-Fights Dataset
Today plus 1 week	Run pre-processing code to decompose surveillance feed into individual frames marked as violent or non-violent through entire dataset.	
Today plus 2 weeks	Review OpenPose code and determine best statistical representation for poses.	

Reflection

The quality of videos in the NTU CCTV-Fights dataset is impressive. In total, the dataset has many hours of real CCTV footage from Singapore and mobile recording footage of fights as well. Having access to this dataset is awesome and I can't wait to work with it in detail. Running my pre-processing code through a segment of about 100 videos (0.3 GB total) took around 30 minutes to finish, so I am somewhat concerned about the length of time it might take for the code to run through the other 16 GB. Hopefully, I won't need a GPU yet and running this code overnight with checkpoints will be sufficient. Below I've added a code snippet that outlines the features I am keeping track and parts of my for loop.

```
for video in VIDEOS:
    start = time.time()
    print("----- VIDEO: ",video," -----")

    index = video.split(".")[0]
    fight_ground_truth = ground_truth["database"][index]
    fps = fight_ground_truth["frame_rate"]
    num_frames = fight_ground_truth["nb_frames"]
    isCCTV = True if fight_ground_truth["source"] == "CCTV" else False

    fights = []
    for a in fight_ground_truth["annotations"]:...

    cap = cv2.VideoCapture(os.path.join(ROOT,"ROSE_701-800",video))
    if (cap.isOpened() == False):
        print("Error opening video stream or file")
    while (cap.isOpened()):
        # Capture frame-by-frame
        ret, frame = cap.read()
        print("CURRENT FRAME ", len(frames))
```