

Daily Log

Tuesday October 15

Researched potential architectures for machine learning model. Last year, for our science fair project, we used a Support Vector Machine to classify poses at a single point in time. One thing that I started to realize is that the model may perform better if it has a spatio-temporal account of the situation. There's very little work that's been done on that specifically in the literature (in the context of violence detection), so I started out by researching existing spatio-temporal models.

Thursday October 17

Researched specific architectures that could account for spatio-temporal information. I found this paper especially interesting: https://milets19.github.io/papers/milets19_paper.8.pdf. The classic problem in creating spatio-temporal models is coming up with a way to translate spatial features to input data that can be easily learned by the network. A graph obviously is a good, simple representation of spatial elements, and could be used to model poses as well (limbs are edges, joints are vertices). The paper is interesting because it calculates an attention score as a representation of different features in the graph, which I don't yet completely understand but am trying to since it could be interesting. Obviously, having a RNN helps take into account time. Our Ramen Party today was also unsuccessful.

Monday October 21

Continued writing code that runs OpenPose across existing frames and generates a label of violent or non-violent. I'm not sure about some parts of this code because it might depend on the architecture I choose to use for the actual model (i.e. might be useful to organize it into timesteps if I choose to use a RNN, etc.) so will probably hold off on this as I continue researching potential architectures. While reading some previous literature on violence detection, I found a paper that has a temporal element, but only across two consecutive frames.

Tuesday October 22

One of the problems I thought of related to having something like a RNN be my architecture could be the number of samples necessary for the model (in real time) before it can detect violence. This might make its use in the real world somewhat impractical. Thus, the architecture I use must be flexible enough to somehow tell whether or not a violent situation is occurring at a specific time but then also account for previous and future frames. I tried researching that but didn't find anything great. Also, I started re-thinking my representations of OpenPose output: if I were

to choose a model with some structural features of graphs as input data, I would need to find corresponding parts of the OpenPose code there.

Thursday October 24

Refreshed myself on the math behind RNNs, vanishing gradient problem, LSTMs, etc. Do think this would be a good architecture, but I'm not sure if it's going to be memory efficient – running OpenPose itself takes a super long time, and training some sort of RNN network on graphs across a huge number of frames could take a really long time and be inefficient. Also started thinking about how the network could be constructed so that it can provide information as to whether or not a violent situation is occurring at a specific point. Successful Ramen Party.

Timeline

Date	Goal	Met
Today minus 2 weeks	Review OpenPose code and determine best statistical representation for poses.	Completed
Today minus 1 week	Run OpenPose code through dataset of frames marked as violent or non-violent.	Paused - want to work on finding best architecture first
Today	Come up with architecture for machine learning system	Still in progress.
Today plus 1 week	Write code that loads new dataset, preprocesses it, and gets it ready as input data for the network	
Today plus 2 weeks	Run network, refine model.	

Reflection

Sort of took a step back from writing code to think about the overarching architecture I was planning on using. I think having an RNN with a temporal sense of how people's poses change over time will be really interesting to see, especially since nobody's done that before in the literature. Kind of unsure about the specifics of how to implement what I'm thinking about in Keras, but I'm sure I can figure that along the way and learn some new things.