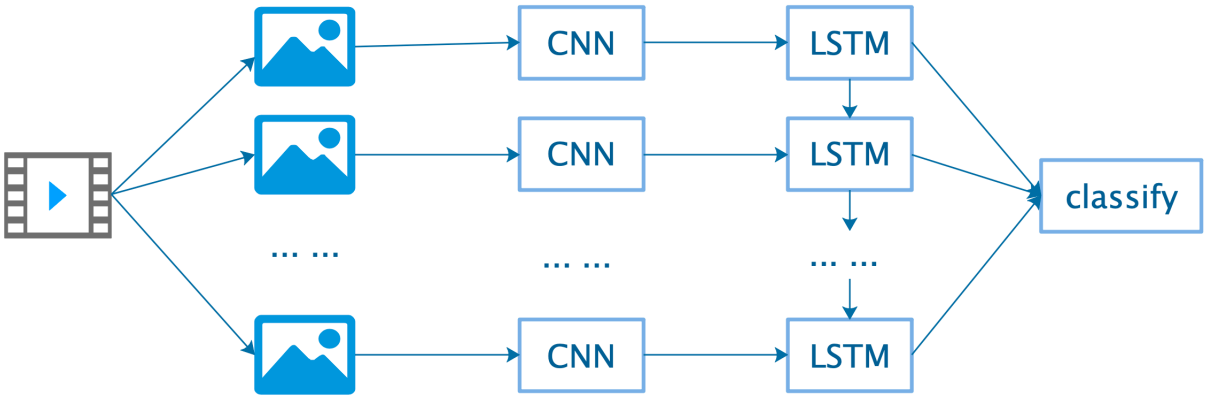
CSCE-636-Project-Part 2

Intuitively, since our task is to detect a **slipping** action in a clip of video, I think it should be a good choice to combine CNN and LSTM to solve this task: using CNN to extract the features in the space aspect for every frame from the video sequences, then using LSTM to extract the features in the time aspect and output the classification results.

Model

We can first cut the video into several frames and each frame is a single image. (we can also add skeleton data for our original videos and it must be a good way to improve the performance of our model.) Then we can feed the images into CNNs and get the features of the images after some rounds of convolution and max pooling. Then we encode the features into some vectors and feed them into LSTMs one by one. Finally we use the output of LSTMs to do a binary classification just like what we did in the IMDB example. (I have read some papers about this approach, some of them choose to average the output of LSTM cells and take the maximum as the output for the LSTMs.)

It shows the figure below:



Input and Output

The input should be some videos with skeleton data and a tag of whether it is a **slipping** action. And For the videos with different lengths, we need to normalize them beforehand. For example, maybe we only decide to use 600 frames in our model, but our input videos may have 100,000 frames and only a short clip of this video contains the action "slipping". In this case, we need to cut out the period with the "slipping" action from the original video. For the other case in which the video is too short, we may insert some empty frames as the padding.

The output is a probability of whether it is a **slipping** action. We may set a threshold to do the binary classification.

Datasets

For the dataset, I plan to find some clips of videos that have **slipping** action inside. Then combine them with some videos which don't have **slipping** actions inside from some open action CV datasets like UCF101, HACS. And make the scales of each type of videos be balanced (like 50% to 50%).

To keep our task simple, I will also remove the videos in which there're over 1 people from our dataset.