CSCE689 Project

Name: Pei Chen UIN: 130005135

Main Contributions

1. The final model structure has two components: Frame Feature Extraction Layer + Sequence Labeling Layer. The first component is to capture spatial features of each frame and the second component is to capture temporal relations among sequential frames.
2. I choose 3D ResNet 34 as the Frame Feature extractor and a BiLSTM layer as the Sequence Labeling layer in the final setting. I also tried other feature extraction components (like 2D pretrained ResNet or common CNNs) and sequential labeling models (like CRF, Transformers), But they do not perform better than the model described in my final setting.
3. Further enlarge the datasets (41 videos) of the 3 types of actions to include more instances in diversified situations and backgrounds.
4. Set annotations rules to ensure annotation quality of the datasets.
5. Add negative examples of the 3 types of actions intertwinedly to maintain the stability of the model, unless the model will be subject to false positives.
6. Use body mark video only (so the input videos only include the skeleton of the human activities) to ensure that the model will work regardless the backgrounds or who does the actions.
7. While training, random sample L (16 or 8) sequential frames as a clip for the 3DResNet34+BiLSTM training. This random segmentation method works much better than uniformly sampling from training videos;
8. Penalize loss function for the imbalanced positive and negative training samples to improve performance.
9. I also tried other ideas like data augmentation methods (e.g. random rotate the frames), differentiate each frame with its previous frame or the first frame in the clips, but it seems they are not helping.

Notes for Testing Your Videos

1. Please use the videos that ***only include body marks***! Which means, use parameter "--disable_blending" when adding body mark to original videos in OpenPose.
2. Please ensure that the input videos are recorded with ***vertical screen mode*** (which means for each frame, Height > Width), or the model will not perform well because the dada preprocessing only support vertical videos.
3. If the test video is very long, please run on CPU or the memory of GPU will not be sufficient.
4. Put your videos in the "marked_only_videos" directory for each type. For example, when wanting to test a video *sample.markonly.mp4* with foot pacing model, put it into the directory ***"./pacing/marked_only_videos/sample.markonly.mp4".***
5. After testing, you will find a ***.png*** figure within the directory.

Research Topic

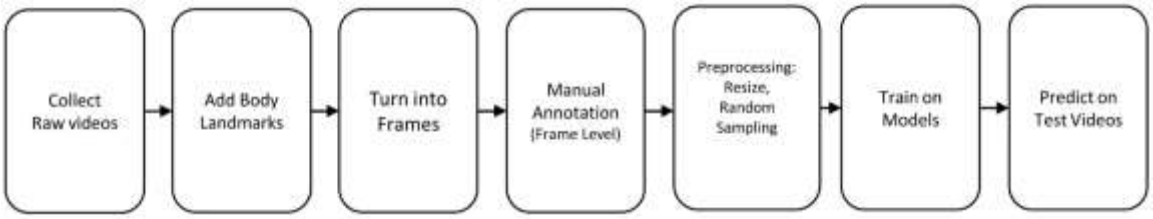
Detecting foot sentiment in videos. Here are 3 types of foot sentiment.

Nervous Pacing: Many people will pace when they are stressed. This acts as a pacifier, as all repetitive behaviors do.

Foot withdrawing: During situations like interviews, interviewees will suddenly withdraw their feet and tuck them in under their chains when they are asked sensitive questions they might not like.

Foot turning away: When we are talking to someone, we might signal that we need to leave by gradually or suddenly pointing one foot toward the door. This is our non-verbal way of communicating "I have to go."

Procedures



1. Collet videos from YouTube, Self-recoding and trim them into clips.
2. Use *OpenPose* tools mark “Body Landmarks” in all clips and only keep the body marks;
3. Then transform the videos into frames, and store each frame as an image;
4. After that, manually annotate each frame as Positive (with target action) or Negative (without target action);
5. Data preprocessing: Resize each frame into 480*272 (H*W) size; Add more negative samples from other types of videos to increase robustness, for example, use foot pacing clips as negative samples for foot withdrawing detection task. I also use a random sampling procedure to get training frame sequences.
6. Train separate models on the 3DResNet+BiLSTM for each action type to classify the frames;
7. Test on the corresponding test data and get the scores for each frame.

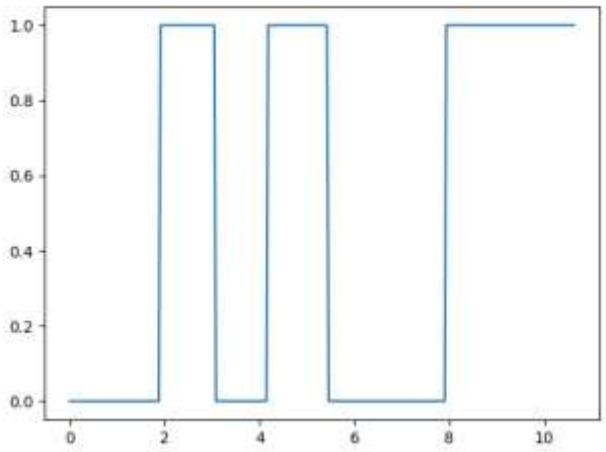
Dataset

Data Size I collect 41 videos clips for all the 3 types of actions under different situations for both training (32) and developing (9). Each clip is about 2-10 seconds and marked "Body Landmarks" using *OpenPose*. I enable the parameter "--disable_blending" when adding body mark to original videos in OpenPose so in all the datasets (either training, developing or blind testing data) only include human skeleton in them.

	Training Clips Count	Developing Clips Count
Nervous pacing	6	2
Foot withdrawing	12	4
Foot turning away	14	3
Total	32	9

Annotation Rules In order to ensure the quality of the annotation process, I set the following rules to annotate each clips.

1. For Nervous Pacing, only the frames in which the subject **is pacing and two legs are both visible** are annotated as Positive frames, otherwise the frames will be annotated as Negative. Take the *foot_pacing_trian_1.markonly.mp4* as an example, the subject is pacing into the frame and the 2 legs are visible starting at about 2nd second and then the frames are annotated as Positive since then. Meanwhile, the subject stops pacing and turning around at about 3rd second then frames are annotated back to Negative. After turning around, the subject starts pacing again about 4th second and Positives are annotated. But the subject is pacing out of the video starting from about 5th second so Negative frames are annotated then. Soon after staring at 8th second, the subject is pacing again into the frame, so Positives are annotated again.



foot_pacing_train_1_gold_label

2. For foot withdrawing, Positive frames are annotated only when one or both of the legs of the subject start to withdraw. When both of the legs are not withdrawing

anymore, Negatives are annotated. The withdrawing procedure usually last about 30-50 frames.

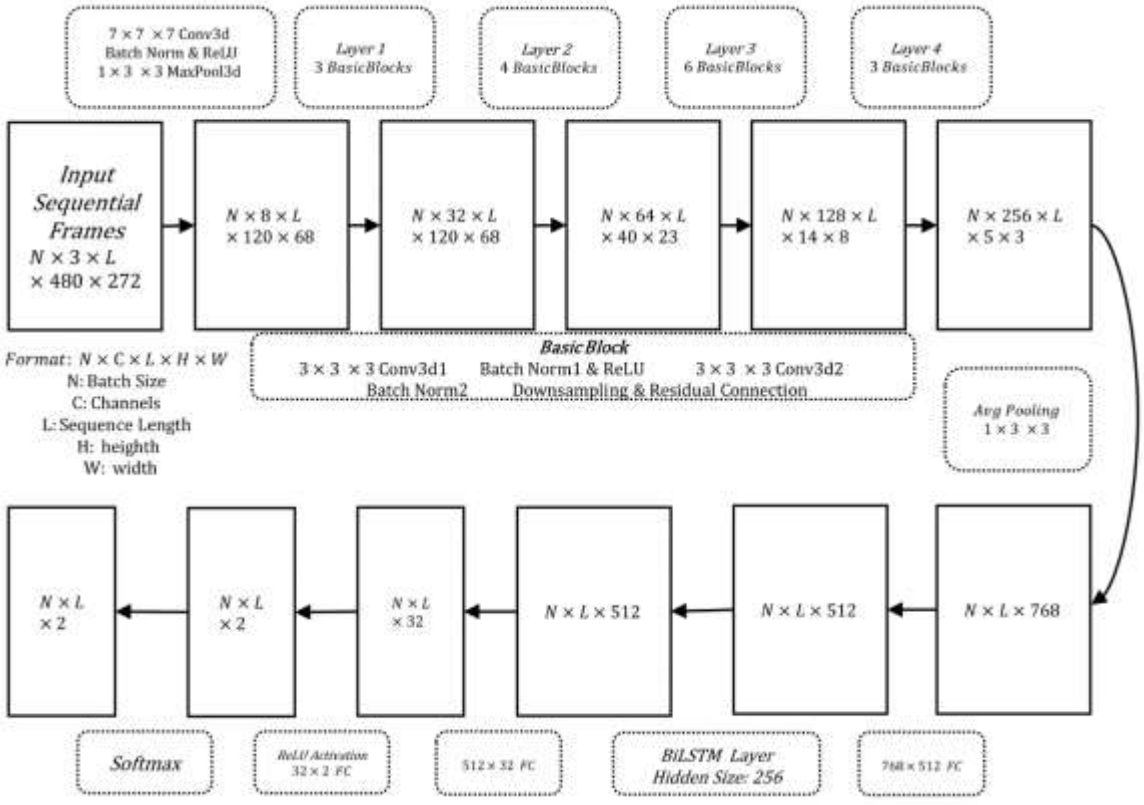
3. For foot turning away, Positive frames are annotated only when one the legs of the subject starts to turn. When the turning stops (toes are not moving anymore), Negatives are annotated. The turning process usually last about 15-30 frames so this action detection is very sparse.

Data Splitting Because I add negative examples of the 3 types of actions intertwinedly, so the final *train/dev/test* splitting of the dataset is as following (All the evaluation performance is based on this splitting)

<table>
<tr><th colspan="2"></th><th>Training Data</th><th>Developing Data</th></tr>
<tr><td rowspan="3">Nervous pacing</td><td>Positive Clips</td><td>foot_pacing_trian_1.markonly.mp4
foot_pacing_trian_2.markonly.mp4
foot_pacing_trian_3.markonly.mp4
foot_pacing_trian_4.markonly.mp4
foot_pacing_trian_5.markonly.mp4
foot_pacing_trian_6.markonly.mp4</td><td>foot_pacing_test_1.markonly.mp4
foot_pacing_test_2.markonly.mp4</td></tr>
<tr><td rowspan="2">Negative Clips From Other Types</td><td>foot_withdrawing_trian_1.markonly.mp4
foot_withdrawing_trian_2.markonly.mp4
foot_withdrawing_trian_3.markonly.mp4
foot_withdrawing_trian_7.markonly.mp4
foot_withdrawing_trian_10.markonly.mp4
foot_withdrawing_trian_11.markonly.mp4
foot_withdrawing_trian_12.markonly.mp4</td><td>foot_withdrawing_test_1.markonly.mp4
foot_withdrawing_test_2.markonly.mp4
foot_withdrawing_test_3.markonly.mp4</td></tr>
<tr><td>foot_turning_away_train_1.markonly.mp4
foot_turning_away_train_4.markonly.mp4
foot_turning_away_train_7.markonly.mp4
foot_turning_away_train_8.markonly.mp4</td><td>foot_turning_away_test_1.markonly.mp4
foot_turning_away_test_2.markonly.mp4</td></tr>
<tr><td rowspan="3">Foot withdrawing</td><td>Positive Clips</td><td>foot_withdrawing_trian_1.markonly.mp4
foot_withdrawing_trian_2.markonly.mp4
foot_withdrawing_trian_3.markonly.mp4
foot_withdrawing_trian_4.markonly.mp4
foot_withdrawing_trian_5.markonly.mp4
foot_withdrawing_trian_6.markonly.mp4
foot_withdrawing_trian_7.markonly.mp4
foot_withdrawing_trian_8.markonly.mp4
foot_withdrawing_trian_9.markonly.mp4
foot_withdrawing_trian_10.markonly.mp4
foot_withdrawing_trian_11.markonly.mp4
foot_withdrawing_trian_12.markonly.mp4</td><td>foot_withdrawing_test_1.markonly.mp4
foot_withdrawing_test_2.markonly.mp4
foot_withdrawing_test_3.markonly.mp4
foot_withdrawing_test_4.markonly.mp4</td></tr>
<tr><td rowspan="2">Negative Clips From Other Types</td><td>foot_pacing_trian_1.markonly.mp4
foot_pacing_trian_2.markonly.mp4
foot_pacing_trian_3.markonly.mp4</td><td>foot_pacing_test_1.markonly.mp4
foot_pacing_test_2.markonly.mp4</td></tr>
<tr><td>foot_turning_away_train_1.markonly.mp4
foot_turning_away_train_4.markonly.mp4
foot_turning_away_train_7.markonly.mp4
foot_turning_away_train_8.markonly.mp4</td><td>foot_turning_away_test_1.markonly.mp4
foot_turning_away_test_2.markonly.mp4</td></tr>
<tr><td>Foot turning away</td><td>Positive Clips</td><td>foot_turning_away_train_1.markonly.mp4
foot_turning_away_train_2.markonly.mp4
foot_turning_away_train_3.markonly.mp4
foot_turning_away_train_4.markonly.mp4
foot_turning_away_train_5.markonly.mp4</td><td>foot_turning_away_test_1.markonly.mp4
foot_turning_away_test_2.markonly.mp4
foot_turning_away_test_3.markonly.mp4</td></tr>
</table>

		foot_turning_away_train_6.markonly.mp4 foot_turning_away_train_7.markonly.mp4 foot_turning_away_train_8.markonly.mp4 foot_turning_away_train_9.markonly.mp4 foot_turning_away_train_10.markonly.mp4 foot_turning_away_train_11.markonly.mp4 foot_turning_away_train_12.markonly.mp4 foot_turning_away_train_13.markonly.mp4 foot_turning_away_train_14.markonly.mp4	
	Negative Clips From Other Types	foot_pacing_trian_1.markonly.mp4 foot_pacing_trian_2.markonly.mp4 foot_pacing_trian_3.markonly.mp4	foot_pacing_test_1.markonly.mp4 foot_pacing_test_2.markonly.mp4
		foot_withdrawing_trian_1.markonly.mp4 foot_withdrawing_trian_2.markonly.mp4 foot_withdrawing_trian_3.markonly.mp4 foot_withdrawing_trian_7.markonly.mp4 foot_withdrawing_trian_10.markonly.mp4 foot_withdrawing_trian_11.markonly.mp4 foot_withdrawing_trian_12.markonly.mp4	foot_withdrawing_test_1.markonly.mp4 foot_withdrawing_test_2.markonly.mp4 foot_withdrawing_test_3.markonly.mp4
Blind Test Video	2600_1.markonly.mp4, 2600_2.markonly.mp4, 2600_3.markonly.mp4, 2600_4.markonly.mp4 2700_1.markonly.mp4, 2700_2.markonly.mp4, 2700_3.markonly.mp4, 2700_4.markonly.mp4 4400_1.markonly.mp4, 4400_2.markonly.mp4, 4400_3.markonly.mp4, 4400_4.markonly.mp4		

Architecture:



The final model structure has two components: Frame Feature Extraction Layer + Sequence Labeling Layer. The first component is to capture spatial features of each frame and the second component is to capture temporal relations among sequential frames.

I choose 3D ResNet 34 as the Frame Feature extractor and a BiLSTM layer as the Sequence Labeling layer in the final setting. As in the figure, the 3D ResNet34 mainly contains a $7 \times 7 \times 7$ 3d Convolutional Layer (with max pooling) and 4 layers residual layers. Within each residual layer, two $3 \times 3 \times 2$ 3d convolutions (with batch normalization and ReLU activation) and residual connections are used. After it, a global average pooling layer and a fully connected layer are employed. Then I feed the feature enriched sequence of frame into a BiLSTM layer to capture temporal relations among the frames. Note that the 3D ResNet does not damage the temporal order of the frames. After that, we use two fully connected layers and a softmax function to predict each frame as Positive or Negative.

Historically, I also tried other feature extraction components (like 2D pretrained ResNet or common CNNs) and sequential labeling models (like CRF, Transformers), But they do not perform better than the model described in my final setting.

Hyperparameters

The main parameters the 3 models are the same as followed. I did enormous of parameter tuning and these settings perform best.

	Nervous Pacing	Foot Withdrawing	Foot Turning Away
Optimizer	Adam		
Max Epochs	50		
Batch Size	8	9	14
Learning Rate	0.0005	0.001	0.0005
Sampling Rate	16	8	8
Loss Penalty (Pos vs. Neg)	3:1	17:3	22:3

Sampling Rate During training, I use a segment sampling method to generate clips which have L frames in it, an L is the Sampling Rate. In detail, I randomly generate an integer (between 1 and Number of frames of a video – L) and take the next L frames as one training sample. There are **three main benefits** to do so: The first one is to control the memory consuming of the model. Because some of the training clips are too long and feeding the whole sequences of frames is too large for the limited GPU memory; The second benefit is to maintain that all input sequences have the same shape so it will be easy to apply 3D convolutional operations on each sample; The last benefit is to enhance the data diversity because we can generate enormous different sequences from even a single video.

Besides, I find that the sampling rate should be proportional to how long the actions last in a video. For example, foot pacing usually last longer than foot withdrawing, so we should use bigger sampling rate for foot pacing.

Loss Penalty Because the positive and negative frame are not balanced so I use more weight for the fewer positives samples in loss function to help training. The weights are based on the ratio of positive and negative frames in the training data.

Performance

The evaluation performance on my own development set is as followed. P/R/F/Acc means the ***Precision/Recall/F1 Score*** of Positive frames and ***Accuracy*** of both Positive and Negative frames in the developing dataset.

P/R/F/Acc (%)	Foot Pacing	Foot Withdrawing	Foot Turning Away
3DResnet34+Transformer	0/0/0/71	0/0/0/93	0/0/0/95
2DResnet34+BiLSTM	75/**94**/83/89	68/81/74/96	**92**/40/56/97
3DResnet34+BiLSTM	**85**/92/**88/96**	**77/85/81/97**	87/**58/69/98**

Input & Output for Testing

Input Here are some constraints for the input videos so the models can work well**. First**, the input videos should only include body marks! Which means, we should use parameter "--disable_blending" when adding body mark to original videos in OpenPose. **Second,** the input videos should be recorded with vertical screen mode (which means for each frame, Height > Width), or the model will not perform well because the dada preprocessing only support vertical videos.

In order to maintain the consistency for all the clips, I turn all the frames into shape of (480, 272, 3) while preprocessing. Because the image details are not necessary for the action detection, I choose such low definition so as to reduce model settings and memory consuming.

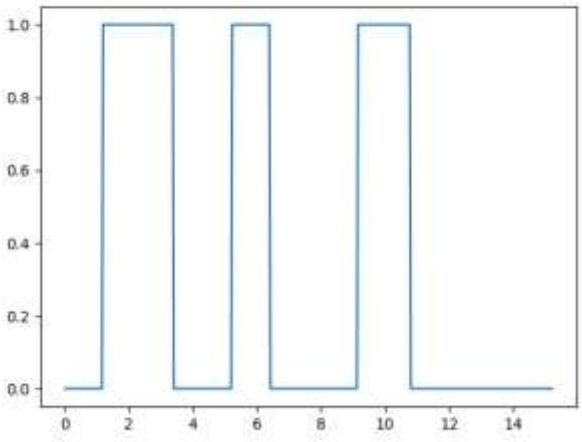
Output For each test clip, I predict the score of each frame with the trained model and store them in a corresponding *JSON* file and a PNG figure.

Result Analysis

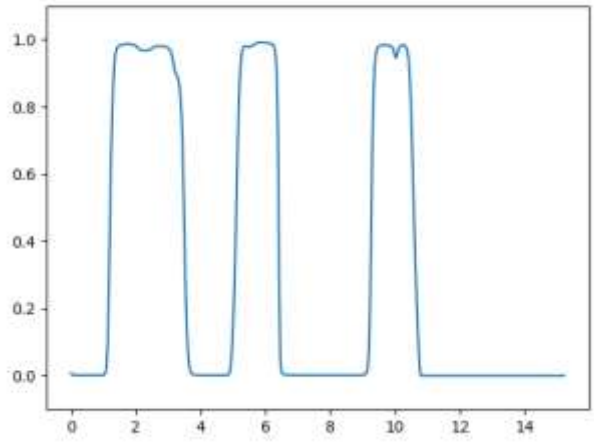
Generally speaking, the models work well in all my own development dataset. And for blind testing videos, my model can successfully identify Foot Pacing actions, and some of the Foot Withdrawing actions, but does not perform as well on Foot Turning Away videos.

Here are some examples of the test figures in from both development and blind test videos.

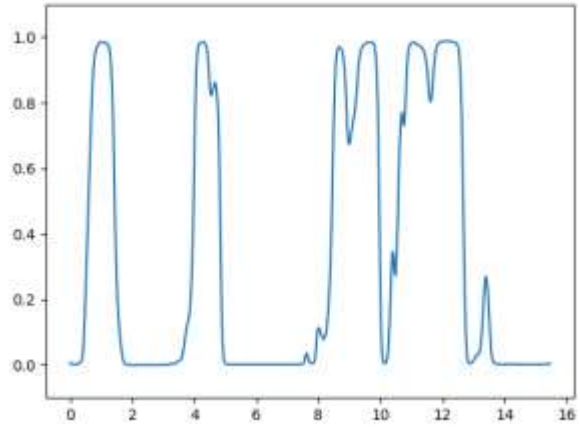
Foot Pacing In foot_pacing_test_1 video, we can see the subject is pacing in and out of the vision alternatively and then sit down on the couch. We can see that the predicted result can successfully identify the start and end of each foot pacing action. Whenever the subject is pacing out of vision or stops pacing, the predicted score will be 0.



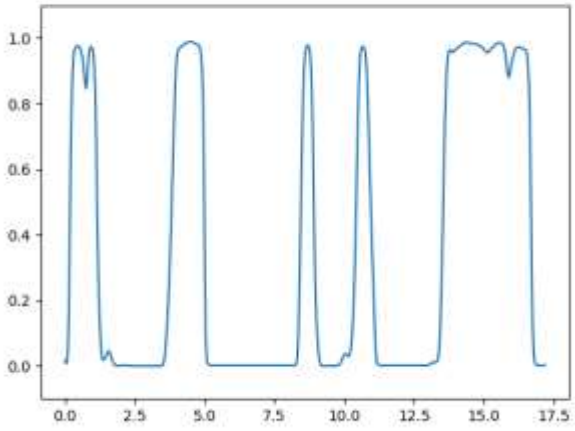
foot_pacing_test_1 gold label



foot_pacing_test_1 predicted label



4400_1 predicted label



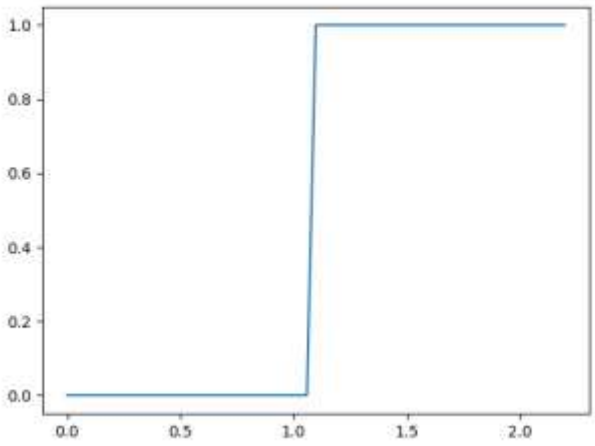
4400_4 predicted label

Foot pacing model also works on the blind test videos. For example, in *4400_1.mp4,* the subject is pacing from the beginning and stops and turns around at about 2nd second. The predicted label shows this trend correctly. Then the subject is pacing again starting from 3rd second and continues to about 5th second. This is also captured by the model. Then the subject turns back into vision staring from about 8th second and turns around again at 10th

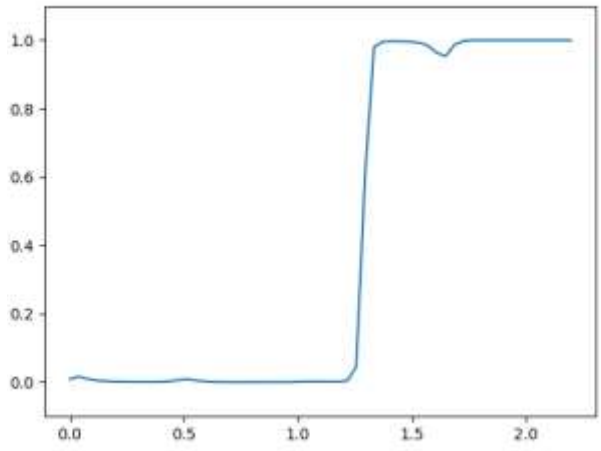
second, and continues pacing until 13th second. All of these details are capture by the model. After that there is only one leg left in the video so the model predicts the frames as negative.

Similarly, in *4400_4.mp4,* the subject is pacing from the beginning and stops and turns around at about 2nd second. The predicted label shows this trend correctly. Then the subject is pacing again starting from 3rd second and continues to about 5th second. This is also captured by the model. Then the subject turns back into vision staring from about 8th second and turns around again at 9th second, and continues pacing until 11th second. All of these details are capture by the model. After that the subject is pacing back at about 13th second and continues until end, which are showed correctly in the predicted label.

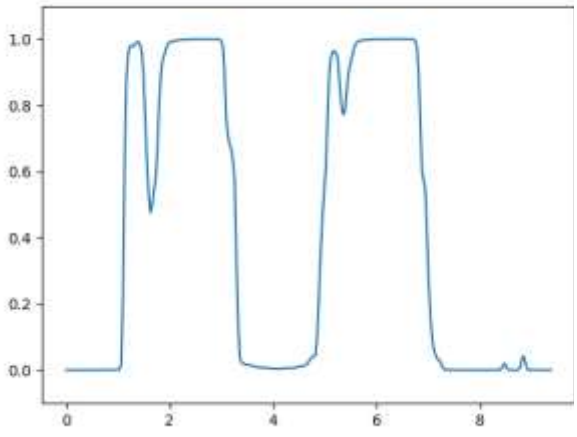
Foot Withdrawing In foot_withdrawing_test_1 video, the subject withdraws feet at about 1st second and continues till end. My model almost predicts this trend perfectly.



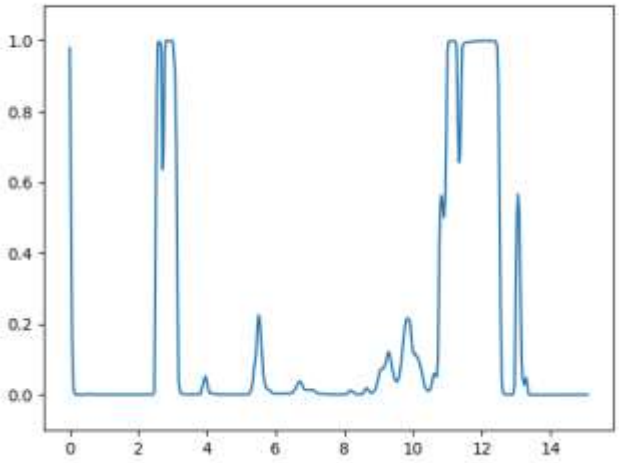
foot_withdrawing_test_1 gold label



foot_withdrawing_test_1 predicted label



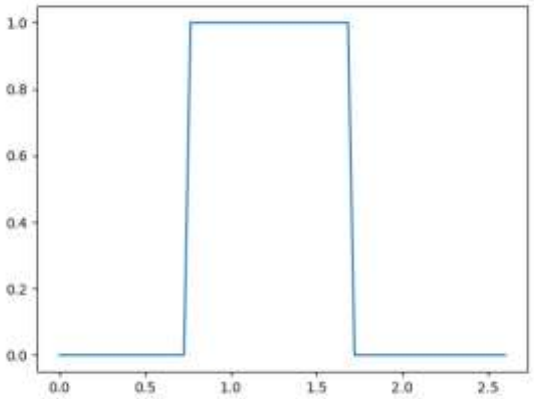
2700_1 predicted label



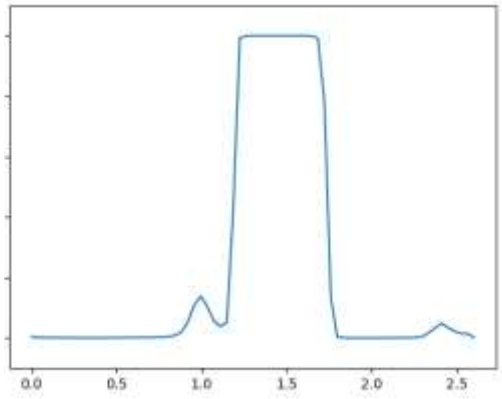
2700_3 predicted label

For blind test videos 2700_1.mp4 and 2700_3.mp4, we can see my model can successfully predict the two foot withdrawing actions during 2700_1 at around 2nd second and 6th second, but only detect one foot withdrawing actions during 2700_3 at around 11th second. It does not identify the foot withdrawing at about 5th second in 2700_3.

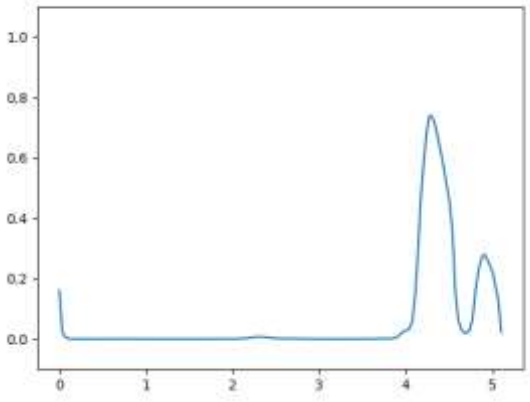
Foot Turning Away In foot_turning_away_test_1 video, the subject withdraws feet at about 0.75 second and continues till 1.75 second. My model almost predicts this action from 1st second to 1.75 second



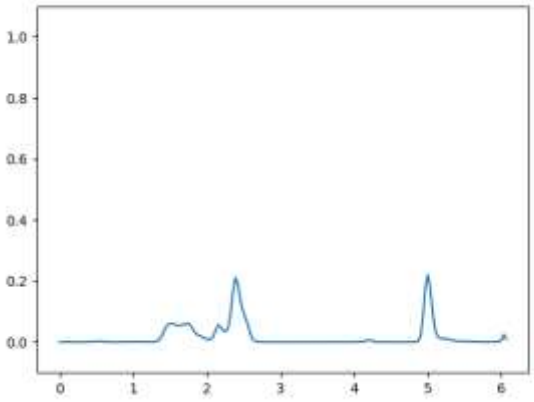
foot_turning_away_test_3 gold label



foot_turning_away_test_3 predicted label



2600_1 predicted label



2600_2 predicted label

But for blind testing videos, the models only predict some clues for foot turning away. As in 2600.mp4, the subject turns foot away starting from 3rd second till end. The model predicts it from 4th second till 5 second. And in 2600_2.mp4, the subject turns foot away from 3rd second to 5th second. Unfortunately, the predicted results only have some higher scores at 3rd and 5ths seconds but fails to predict the time span.

Links

Videos: https://www.youtube.com/playlist?list=PLlUqXrHW9mC_GlGbgiAUmPlwDMZWuLSP4

Code: https://github.com/brickee/foot_sentiment.git