Introducing SSBD+ Dataset with a Convolutional Pipeline for detecting Self-Stimulatory Behaviours in Children using raw videos

Vijay Jaisankar, Vaibhavi Lokegaonkar
Surgical and Assistive Robotics Lab (SARL)
IIIT Bangalore

Overview

- Self-Stimulatory Behaviours
- Our Contributions
  - SSBD+ Dataset
  - Pipeline: M_1 & M_2
- Metrics and Experiments
- Future Work
- Conclusion

What are Self-stimulatory behaviours?

Children diagnosed with an Autism Spectrum Disorder (ASD) condition often perform **self-stimulatory actions** in response to external stimuli, to combat anxiety and stress, etc.
In this work, we consider the following actions:

Armflapping

Headbanging

Spinning

Our Contributions

Our contributions to this work are:

- **SSBD+**: New videos added to original SSBD dataset.
- **SSBDPipeline**: Pipeline-based architecture for identifying self-stimulatory behaviors.

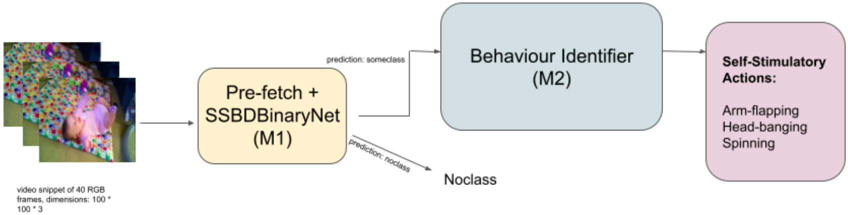
Introducing SSBD+ Dataset

- The SSBD dataset contains video URLs containing timestamps of self-stimulatory actions.
- We augment the dataset into **SSBD+**
- ≈45% more annotated data points available to researchers.

Pipeline: Preprocessing

- **Downsampling** the parent video to 10 fps.
- **Chunking** the parent video into 40 frame chunks.
- **Labelling** the chunks through the sliding window approach.
  - If $\geq 75\%$ of the frames in the chunk are labelled as x where $x \in \{\text{Armflapping, Headbanging, Spinning}\}$, the chunk is labelled as **action** x
  - Else, the video is labeled as **no-class**.

Pipeline of the Classifier Architecture

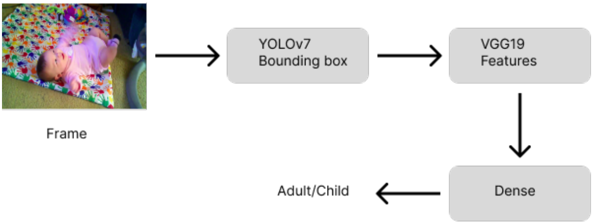


M_1

M_1 : *Detecting the presence of self-stimulatory actions*

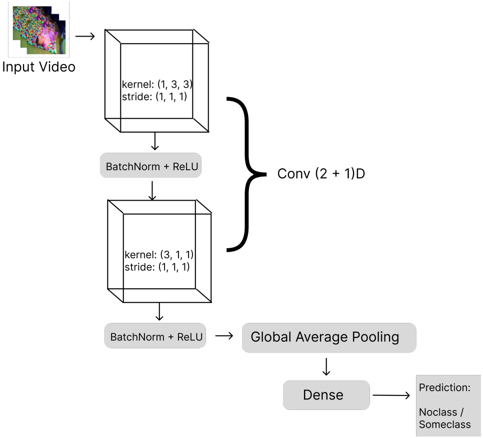
Pipeline Prefetch

As we are focusing on classifying the actions of children, *Prefetch* detects the portion of the video frames containing them using a YOLOv7 + fine-tuned **VGG-19** model.



Pipeline Detector: M_1

M_1 classifies whether the video contains any of the 3 actions by using a **Conv (2+1)D** Architecture.



M_2

M_2 : *Classifying the type of self-stimulatory action into* $\{$*Armflapping, Headbanging, Spinning*$\}$

Frame Selection Algorithm

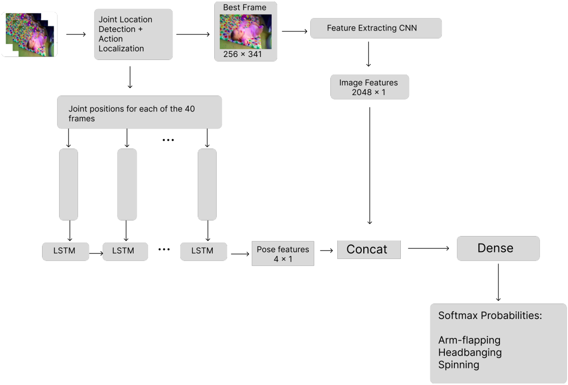
We select the frame with the most difference in joint coordinates with its successive frame and pass that to M_2.

Input: Frames of the single video chunk 1 to 40 in the playing order (F)
Input: Joint coordinates (J) detected in each frame of the chosen video chunk 1 to 40 (in the same order as in F)
Output: Index of the best frame to be evaluated by the model
Initialisation :

```
1: maxDiff = 0
2: maxFrameIdx = −1
3: for t = 1 to t = 39 do
4:     diff = ||J[t] − J[t + 1]||
5:     if maxDiff < diff then
6:         maxDiff = diff
7:         maxFrameIdx = t
8:     end if
9: end for
10: return F[maxFrameIdx]
```

Pipeline Identifier: M_2

M_2 classifies the single video frame along with *Movenet* joint coordinates of all frames into one of the 3 actions present by using a **CNN-LSTM** system.



Discussion

Results, Key takeaways and points, and future work.

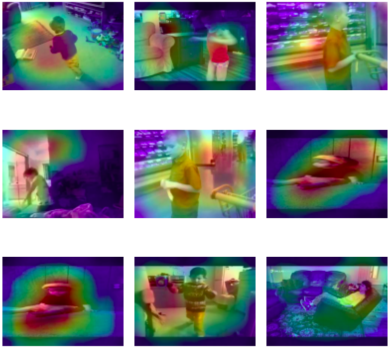
Results: Accuracy and Performance

Table: Accuracy, F1-score, and average FPS of the pipelined models

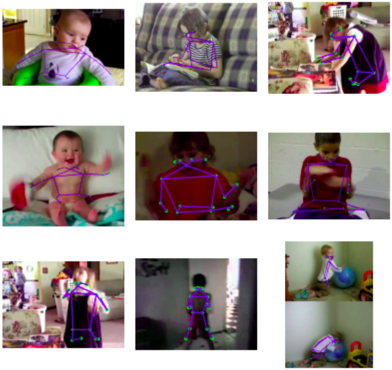
Model	F1-Score	Accuracy	Average FPS
M_1 (with Prefetch)	0.819	0.811	38.265
M_2 (with Frame selection)	0.789	0.812	14.755

Gradcam and Pose Coordinate Images for M_2

Gradcam Images



Pose Coordinate Images



Ablation Study: M_1

- Without child position localization using Prefetch Algorithm
- With child position localization using Prefetch Algorithm – **proposed pipeline**

M_1 **Ablation**	**F1-Score**
Without Prefetch	0.740
With Prefetch	0.819

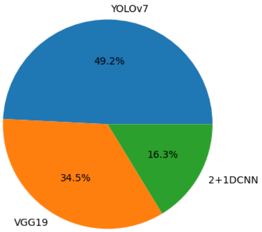
Ablation Study: M_2

- Removal of the Frame selection algorithm in M_2 - using all frames of the video
- Using the Frame selection algorithm in M_2 - **proposed pipeline**

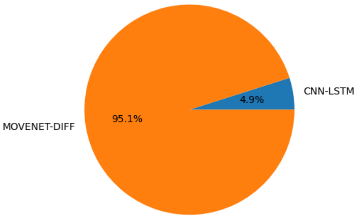
M_2 **Ablation**	**F1-Score**
All Frames	0.652
Single Representative Frame	0.789

Performance: Inference time Breakup

Fraction of inference time taken by elements of the M_1 pipeline



Fraction of inference time taken by elements of the M_2 pipeline



Experiment: Distillation Learning

- **Teacher model**
  - Resnet-18 (*trainable* classifier head) + BiLSTM + Multi-head Attention + 3 Fully-connected layers.
  - *23.8M learnable weights* in M_2 setting.
- **Student model**
  - Resnet-18 (*frozen* classifier head) + LSTM + 2 Fully-connected layers.
  - *8.9M learnable weights* in M_2 setting.

- Loss function of student model
  - L_{CE} - Cross-entropy loss with ground truth labels (weightage: 0.75)
  - L_{SOFT} - Temperature-softened softmax loss with teacher model (weightage: 0.25)
- **Key result**: 37.38% learnable weights and 80.89% relative performance of the teacher model.

Pipeline: Postprocessing

- We pass $k = 2$ video chunks through the pipeline and decide the labels collectively based on the softmax values of each predicted label.
- Addressing the case with M_1 falsely predicting a video containing an action *noclass*: If at least one of the videos is predicted as having one of the actions, we pass both the videos through M_2.
- Addressing the case with M_2 being passed with a *noclass* model: If all the classes have softmax probabilities less than $0.33 + \delta$, the video chunk is labeled *noclass*.

Open-sourced code and data

Project Page

Conclusion

Our contributions to this work are:

- SSBD+: ≈45% New videos added to original SSBD dataset.
- SSBDPipeline: Pipeline-based architecture including prefetch for child coordinates, action detection model (M_1), and action identification model (M_2) for classifying self-stimulatory behaviors.

Discussion Points: Methodology

- **Why not develop a single 4-class classification model?**
  There is a huge imbalance between no-action videos and the videos having some action, i.e. 1 video having action for 7 no-action videos. This leads to a biased model.
- **Why not use an attention-based architecture?**
  Based on our analysis, attention-based architectures achieving comparable results (e.g., *Model distillation experiment*) had considerably larger footprints and were slower to train.
- **How was SSBDPLUS curated?**
  35 new videos gathered from YouTube by searching for the respective actions, for example with the prompt *Headbanging autism actions in children*.

Discussion Points: Performance

Table: Model footprints of various models in the pipeline

Model	Total #Weights	Learnable #Weights
VGG-19 FC: M_1 setting	143.9M	273.4K
2+1D CNN: M_1 setting	38.2K	38.2K
CNN-LSTM: M_2 setting	20.8M	6.7K

Discussion Points: Open problems for future work

- Extending architectures to different actions
  - Tactile actions like *Rocking*
  - Other modalities of self-stimulatory behaviours
- Benchmarking the robustness of such systems to adversarial attacks
- On-device deployment

Thank you

We would love to hear your questions and valuable feedback on this project!