Model Proposal on primitive Classification/Segmentation

Sudharshan Ramesh
Tandon School of Engineering
New York University
New York City, USA
Email: sr7431@nyu.edu

***Abstract*—This report covers on how to classify primitives such as reach, stabilize, transport, reposition and idle and segment them from video data. This analysis was done on top of StrokeRehab: A Benchmark Dataset for Sub-second Action Identification.**

***Index Terms*—StrokeRehab, GNN, Mask, SAM2, TCN, CNN, Primitive**

I. Introduction

Video primitive classification and segmentation involve breaking down complex actions in videos into basic, elemental motions (reach, stabilize, transport, reposition and idle). This is crucial for applications like rehabilitation assessment, gesture recognition, and detailed motion analysis. This report will expand into how to better classify the video segments without using seq2seq or raw2seq models.

II. Related Work

There are several existing bechmarks available that has data for video action identification: Breakfast, 50Salads, Jigsaw, kinetics. But the problem with them is that they have very few short segment actions in the range of seconds or subsecond. The StrokeRehab Dataset used in the **StrokeRehab: A Benchmark Dataset for Sub-second Action Identification** has many sample of subsecond and few second action samples collected from rehab session of stroke-patients and healthy patients.

III. Methodology

The action identification i.e, the primitives can be easily labelled as:

- Reach: Reaching for a target object with the affected arm.
- Stabilize: Picking up the target object with the affected arm and holding it.
- Transport: Move the target object to the action zone and perform desired action.
- Reposition: Move back the target object to the initial position.
- Idle: No movement/interaction on the affected arm.

To proceed with the primitive classification/segmentation, we have to follow the below represented flow:

A. Data Preprocess

The video data has to be split into frames and identify the hands and objects in the frame. Taken the video sample, we first identify the location of the hands in the video. This can be done with help of tools like mediapipe/ Openpose. From the video sample select a single frame having the best confidence. Using this frame as reference get the masks for the hands in the video using SAM2 on both forward and backward iteration of the video frames, concatenating them at the end. Similiarly the object should also be masked and mapped using models like Yolo. Let's see a small comparison for different vision models in the relative usage aspect:

Model	**Accuracy**	**Speed**	**Ease of Use**
MediaPipe Hand	High	Fast (Real-time)	Easy
OpenPose	Very High	Medium (Not real-time)	Hard
OpenCV (Haar)	Medium	Fast (Real-time)	Medium
TensorFlow (HandPose)	High	Medium	Hard
DeepHand (DL)	Very High	Slow	Hard
PoseNet (TF)	High	Medium	Medium

TABLE I
Comparison of hand detection models.

Model	**Accuracy**	**Speed**	**Ease of Use**
YOLOv8	High	Fast (Real-time)	Easy
SSD (Single Shot Detector)	Medium	Fast	Medium
Faster R-CNN	High	Medium	Hard
RetinaNet	High	Medium	Medium
EfficientDet	High	Fast (Real-time)	Easy

TABLE II
Comparison of object detection models.

Model	**Accuracy**	**Speed**	**Ease of Use**
SAM2 (Segmentation)	Very High	Medium	Easy
U-Net	High	Medium	Easy
DeepLabV3+	High	Medium	Medium
Mask R-CNN	High	Medium	Medium
HRNet (Segmentation)	Very High	Slow	Hard

TABLE III
Comparison of image segmentation models with mask application.

B. Feature Extraction

From the listed models we pick mediapipe/ openpose(for hand detection), SAM2(for image segmentation and masking), YOLOv8(for object detection).

Later with the frame based masked data, we need to create graphical nodes representing hands and different objects and edges representing the spacio-temporal features between the objects. The spacio-temporal features can be retrived with the help of a 3D CNN model like X3D or I3D.

Convert the videos frame by frame to grayscale and apply a blur to it to reduce the highlight to the background. Apply image masks on the modified frame and save the new videos, this may help us in bringing the attention to the hands and objects in action. Having the new video data and the same labels as earlier, use a 3D CNN architecture like I3D or X3D to extract spatio-temporal features from the preprocessed video. These models are well-suited for capturing both spatial and temporal information.

C. Model Training

With the graphical nodes data we can train a GNN (Graph Neural Network) model like GCN(Graph Convolutional Networks) or GAT(Graph Attention Networks) to train the labelled graphical nodes.

With the extracted features from the new videos we can use TCN(Temporal Convolutional Network) or transformer based model to model the sequence of primitives.

D. Imporvements

Incorporating mask features: Extract features from the binary masks using a separate 2D CNN and concatenate them with the video features.

IV. Conclusion

On the net having both these models and along with the sensor data, we can propose a multi-modal model, to train on all three features.

References

[1] Aakash Kaku, Kangning Liu, Avinash Parnandi, Haresh Rengaraj Rajamohan, Kannan Venkataramanan, Anita Venkatesan, Audre Wirtanen, Natasha Pandit, Heidi Schambra, Carlos Fernandez-Granda, "StrokeRehab: A Benchmark Dataset for Sub-second Action Identification," *NeurIPS*, 2022.