MAAF
My **A**mazing **A**ssembly **F**urniture

Aurel, Umer, Baturalp, Mahsa
under supervision: M.Sc. Marsil Zakour, Yuankai Wu

17.07.24

Motivation

- Complexity of assembling IKEA furnitures
- Reducing Dependency on Written Manuals
- Improving User Experience
- Addressing Common Mistakes
- Safety Enhancements

Fig.1: Elderly man struggling

Problem Statement

- **Goal**: Develop a FAA by using different fundamental and SOA models
- **Performed**
  - action classification: 6 different classes
  - mistake detection
- **Applied**
  - DEVA: adaptable to different furniture inputs
  - Co-Tracker: adaptable, reduces video feature outputs
  - Pretrained and customized deep learning models
- **Visualization:** 3D reconstruction
  - trimesh
  - pyrender

Dataset Collection

- New Dataset: IKEA Shoe Rack
- Tabletop Sensor Set Up
- 14 different video sequences including some common mıstakes

General Pipeline

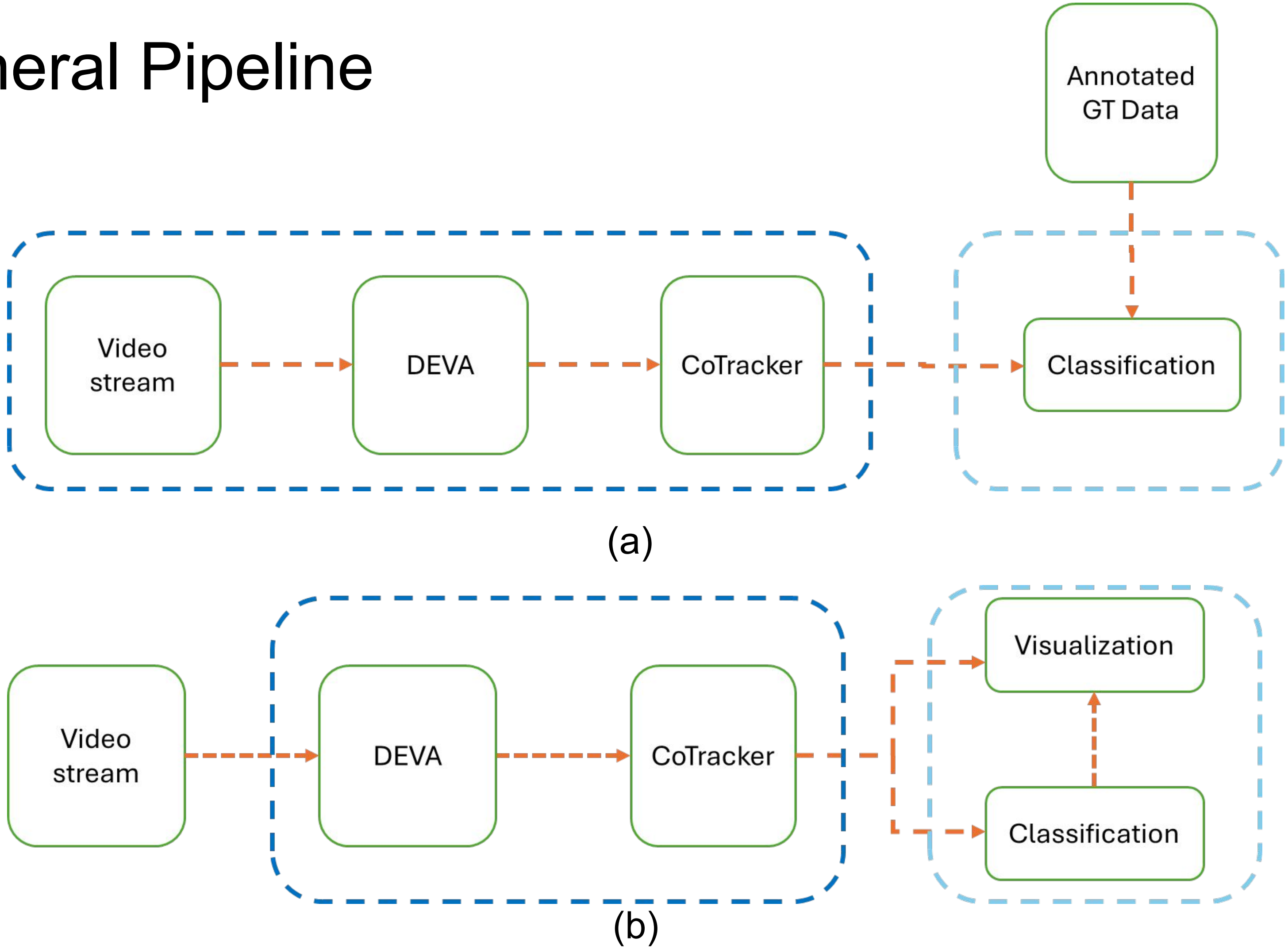


Fig.2: General Pipeline of MAAF (a) Training Pipeline (b) Processing Pipeline

Video Annotation

- VGG Video Annotation: Tool developed by Oxford
- Generate a .json file which includes 6 different action classes and their timestamp for each sequence and view

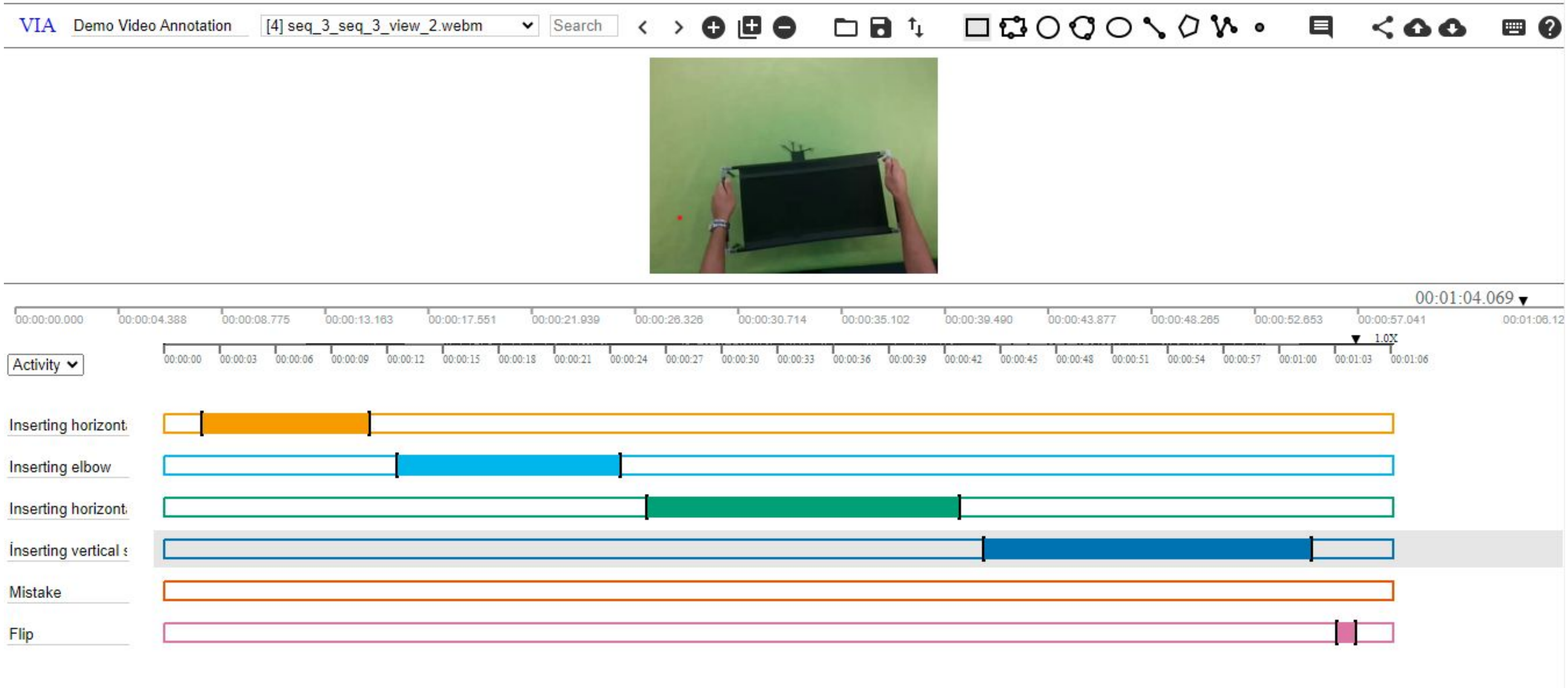


Fig.3: VGG Tool Implementation

Video Annotation

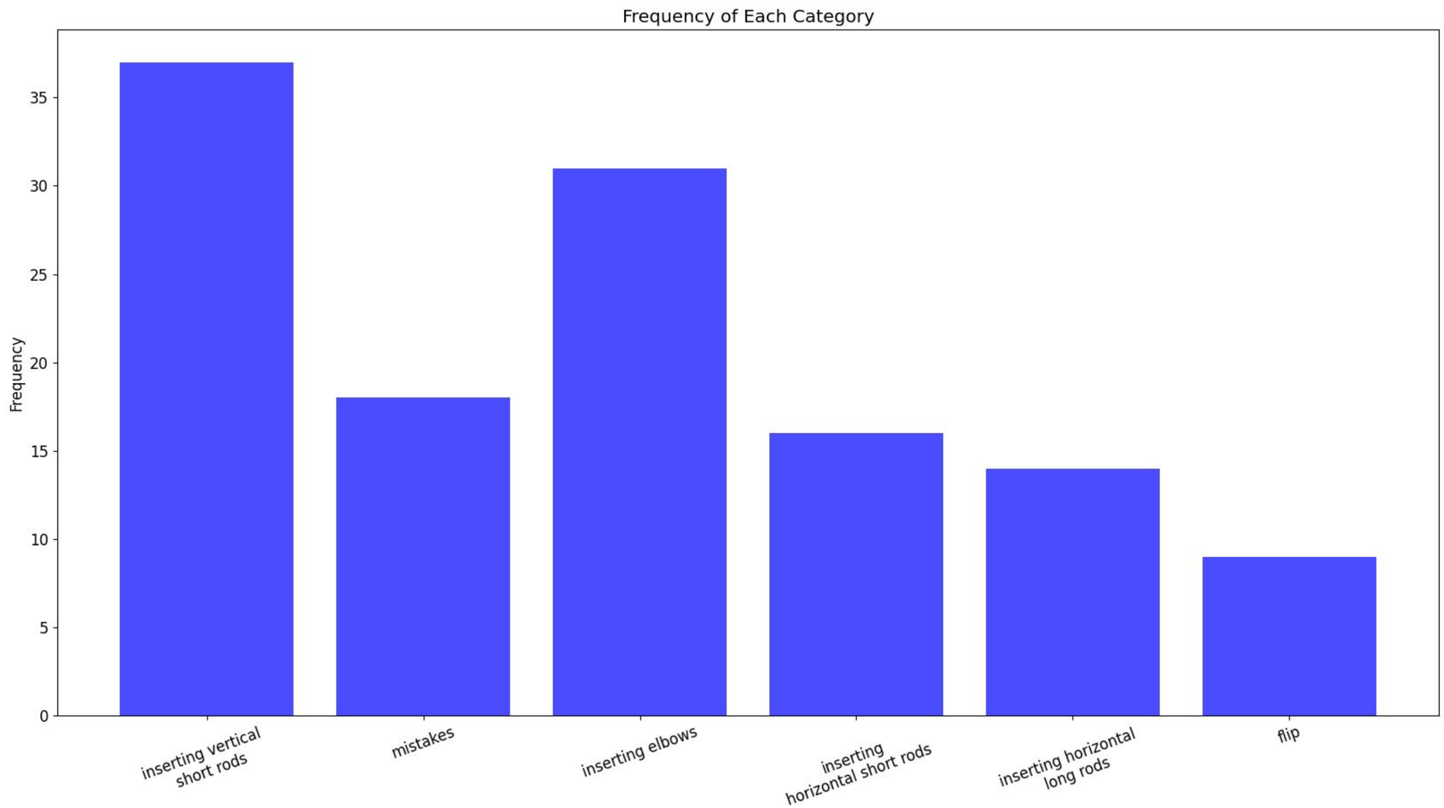


Fig.4: Action classes and their respective frequency

Image Segmentation

1. **Vision**
   a. Utilizing a model that can quickly adapt to segment different objects without fine-tuning
2. **Challenges**
   a. Hard to detect objects
3. **Attainments**
   a. Hands on experience with SoTA segmentation models
   b. Docker and linux file management

DEVA [2]

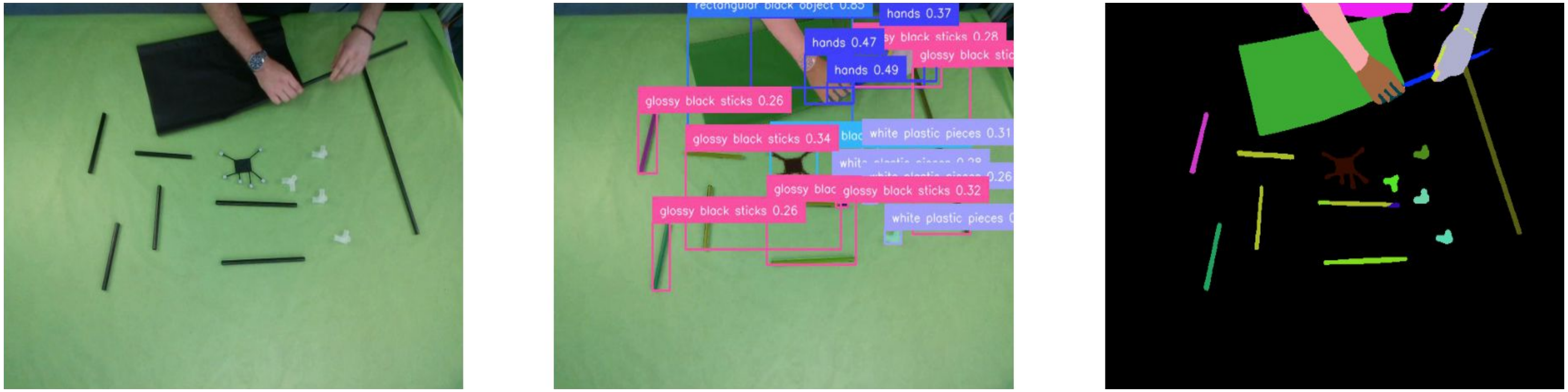


Fig.5: Segmentation Masks

Key Highlights:

- Open-vocabulary zero shot learning model
- Fuses segmentation hypotheses from different frames to generate a coherent segmentation

Advantages:

- Eliminating the need for annotated datasets
- Allows for rapid adaptation in pipelines by changing prompts

Action Classification

1. **Vision**
   a. Taking advantage of multi-view setup
   b. Usage of reduce representation of the scene
2. **Challenges**
   a. Hard to detect objects
   b. Unbalanced and noisy dataset
3. **Attainments**
   a. Hands on experience with state of the art (SoTA) methods
   b. Unbalanced dataset management strategies
   c. Docker and linux file management

Co-tracker [3]

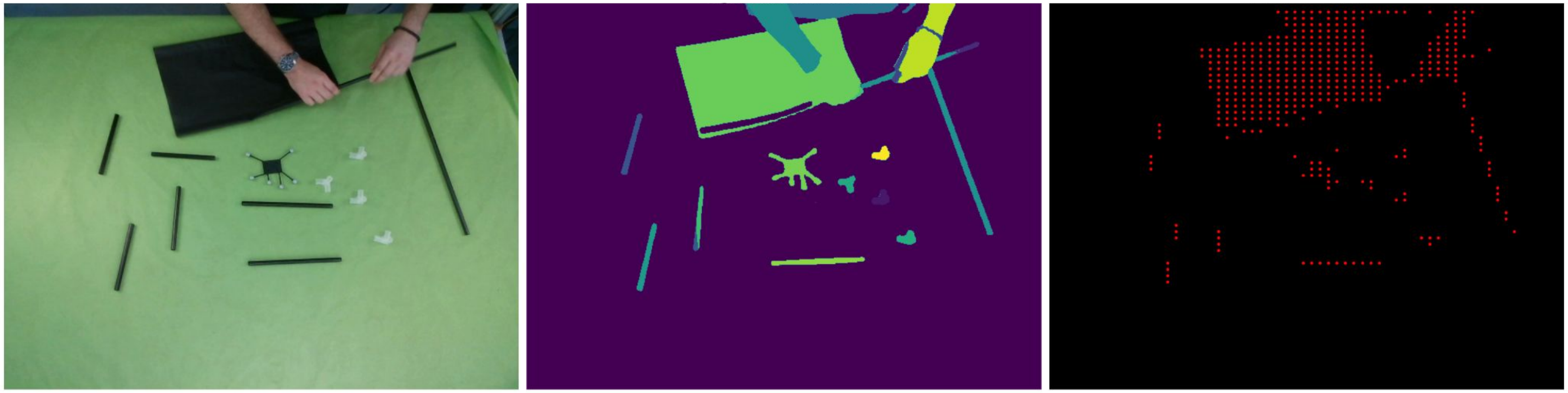


Fig.6: Segmented Co-tracker output

Capabilities:

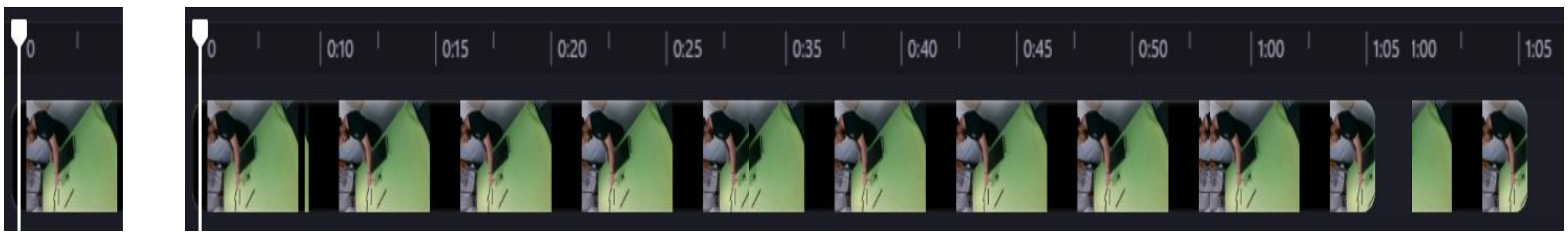
- Transformer based point tracker
- Can detect up to 70K points
- Can detect segmented points

Usage:

- Reduced-feature extraction
- Object Tracking (Visualization)

Dataset Generation

1258374 frames



Problems

- Unequal sequence and views
- Imbalanced dataset
- Low quality labels

Solutions

- Downsampling
- Class weights based on scarcity
- Focus on the only on actions

Methods

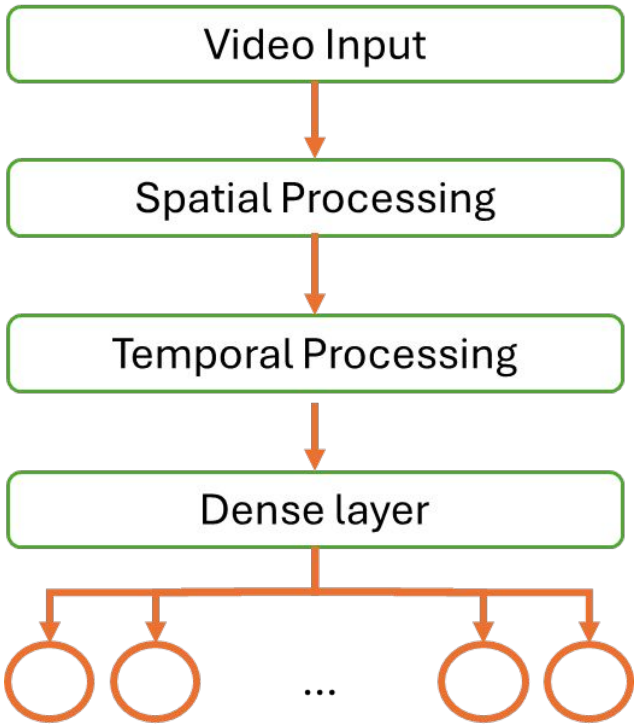


Fig 7: Processing Logic

Comparative Analysis:

Viewpoint	• Multi - Single view
Input modality	• Coordinate - Image
Loss function	• Classical - weighted
Action type	• Action - action+mistake

Architectures

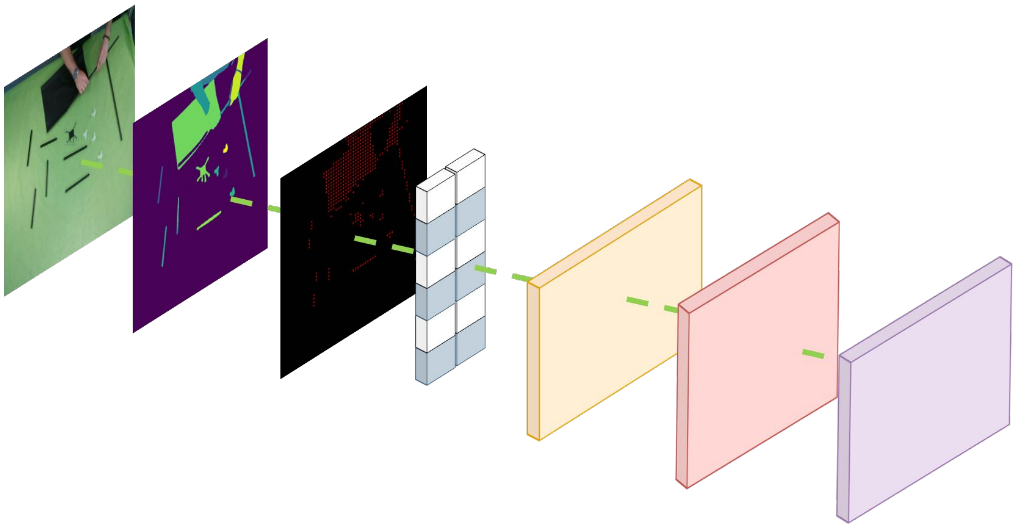


Fig. 8: Custom Lightweight Model: The yellow block represents the Efficient Channel Attention (ECA) [1], the pink block denotes the ReductionCNN block, and the purple block illustrates the ViewAware Transformer.

Ablations on Light Weight Model

Configurations:

- Weighted F1 score
- Warm Up schedule with cosine decay
- 200 epochs

Results:

Table.1: Ablation Table

Model Settings	Highest Val F1 Score
Multiview weighted loss	66.67%
Singleview weighted loss	52.10%
Multiview classical loss	60.87%
Multiview weighted loss with mistakes	44.95%

Results

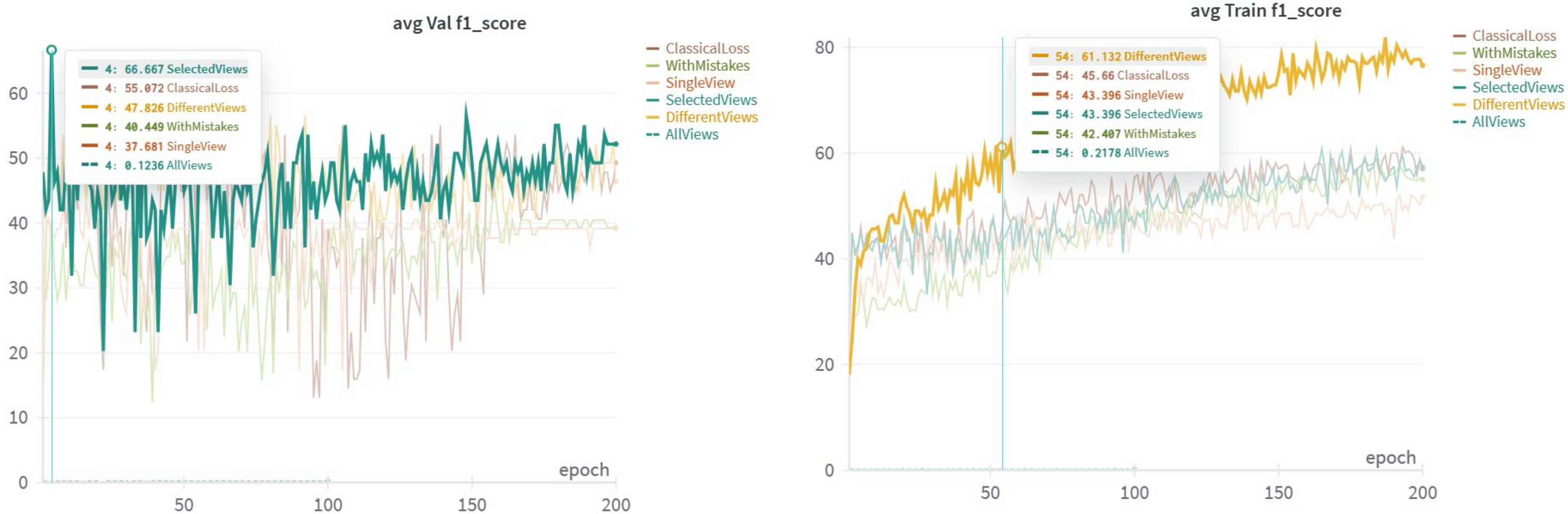


Fig.9 : Ablation graphs

- Multiview performs better
- Selected views effects performance
- Weighted loss performs better
- Accuracy higher without mistakes

ResNet Architecture

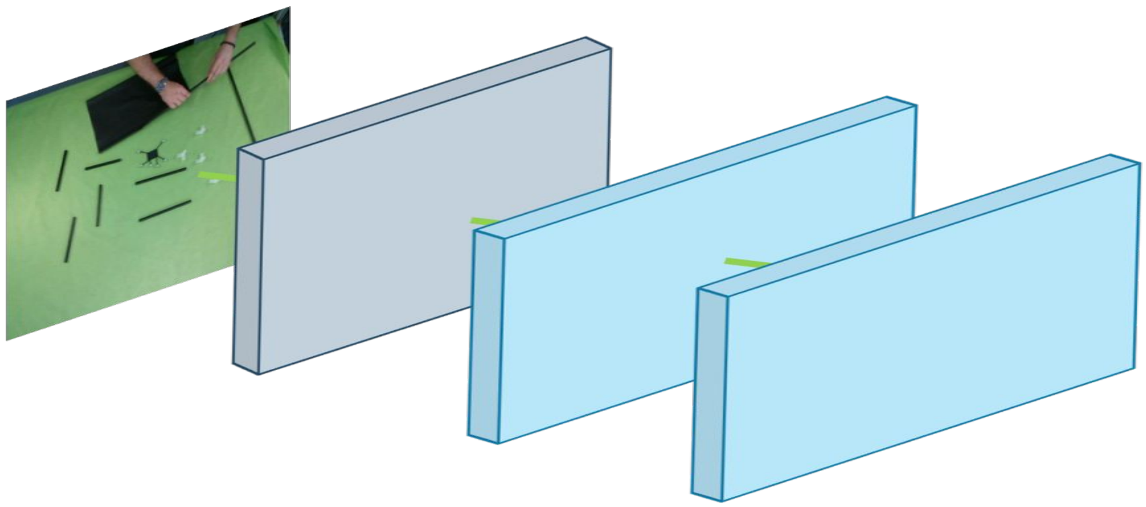


Fig.10: ResNet+LSTM Based Big Model: The gray block represents the ResNet component, while the blue blocks denote the LSTM units.

Ablations on ResNet

Input Ablations on ResNet

- 1 view (5) Resnet-50
- 4 views (2,4,5,6) Resnet-50
- All views Resnet-50

Results:

Table 2: Different input modality performance table

Model Settings	**Highest Val F1 Score**
1 view	85.71%
4 views (2,4,5,6)	76.05%
All views	37.92%

Confusion Matrix for 4 Views on ResNet-50

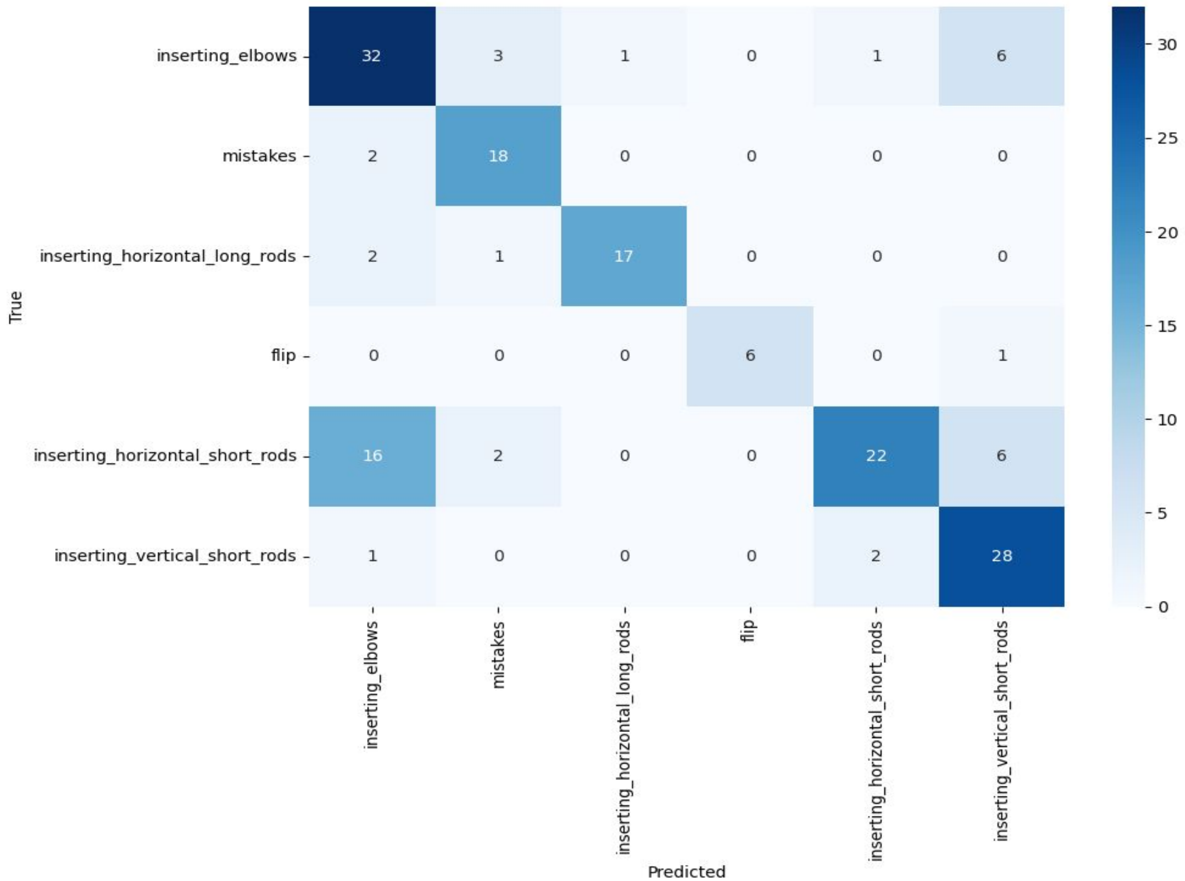


Fig. 10: Confusion Matrix

Input Modalities on ResNet-18

Experiments

- RGB Videos
- Segmentation mask inputs
- Overlayed co-tracker [3] inputs

Results:

Table.3: Different input modality performance table

Model Settings	**Highest Val F1 Score**
Single view classical loss no mistakes rgb input	52.10%
Single view classical loss no mistakes mask input	43.20%
Single view classical loss with mistakes overlay co-tracker input	44.95%

Executive Summary

Implementation Options:

- Lightweight Model:
  - Co-tracker [3] based fast approach
- Big Model:
  - Resnet+lstm based direct rgb approach

Performance Highlights:

- Degradation in the mistake detection
- High accuracy with Resnet+lstm

Table.4: Selected Model Performance

Model Settings	**Highest Val F1 Score**
Resnet+LSTM with Mistakes	76.05%
ECA+CNN+Transformer with Mistakes	44.95%

Demo of Resnet LSTM with Actions

Demo of Resnet LSTM with Mistakes

Demo of Light Weight Model with Actions

Demo of Light Weight Model with Mistakes

Visualization

1) Updating the 3D model of the furniture by receiving the action sequence
2) Movement tracking the components of the furniture

Fig.12 : Final 3D reconstruction

Visualization - 3D reconstruction

Data from the CoTracker

1) Inserting the long horizontal rods
2) Inserting the short horizontal rods
3) Inserting the elbows
4) Inserting the short vertical rods
5) Mistakes

Visualization - 3D reconstruction

1) Trimesh: Constructing the cylinders, spheres, rectangles
2) Pyrender: for rendering and visualization of the meshes

Fig.13: Pyrender’s output

Visualization - 3D reconstruction

General idea:

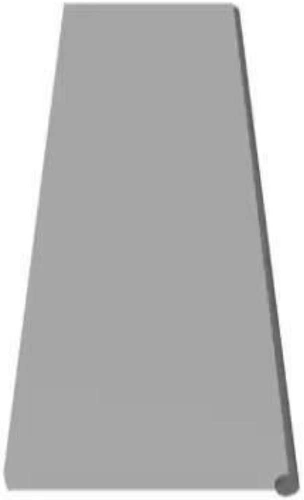
1) Discretely constructing the components
   - Sphere for the elbows
   - Cylinders for the rods
   - Rectangle for the table top
2) Locating them in the desired positions through coordinates
3) Using the timestamps from the pre-trained model

Visualization - 3D reconstruction

Challenges:

1) Differentiating between different rods and elbows
2) Visualization of the mistake due to the orientation
3) Mistake classification

Visualization - 3D Reconstruction



Visualization - Component Tracking

1) Using the segmentation masks to segment different objects
2) Using CoTracker to sample the components
3) Using CoTracker to track the sampled points locations in each sequence
4) Combine the previous steps with 3D-reconstruction

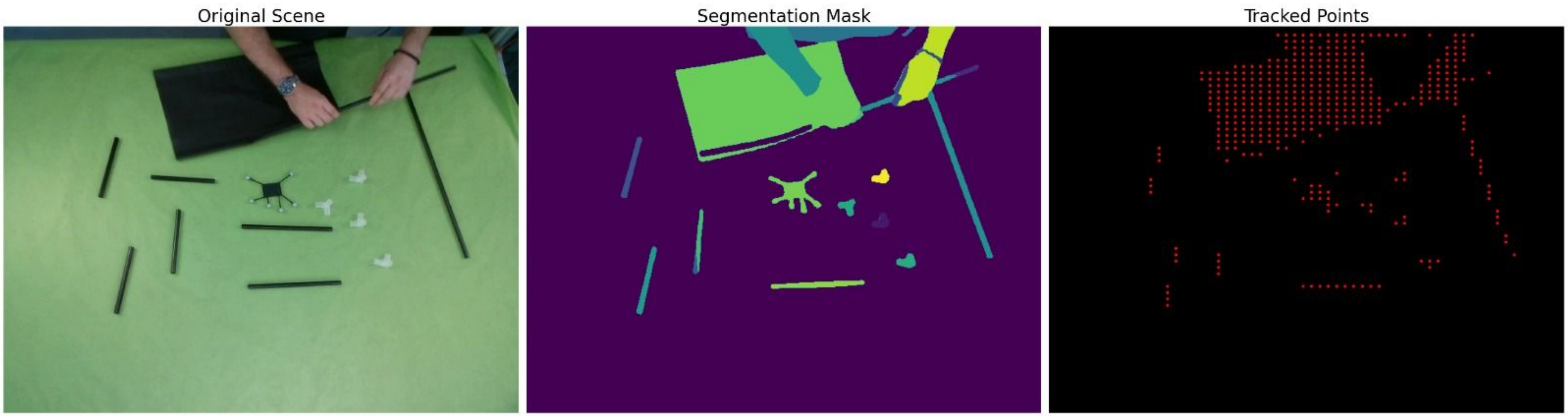


Fig.14: CoTracker’s output

Visualization - Component Tracking

Challenges:

1) Not all the views are useful
2) Some objects are not completely present in the recordings
3) Hand is masking the object

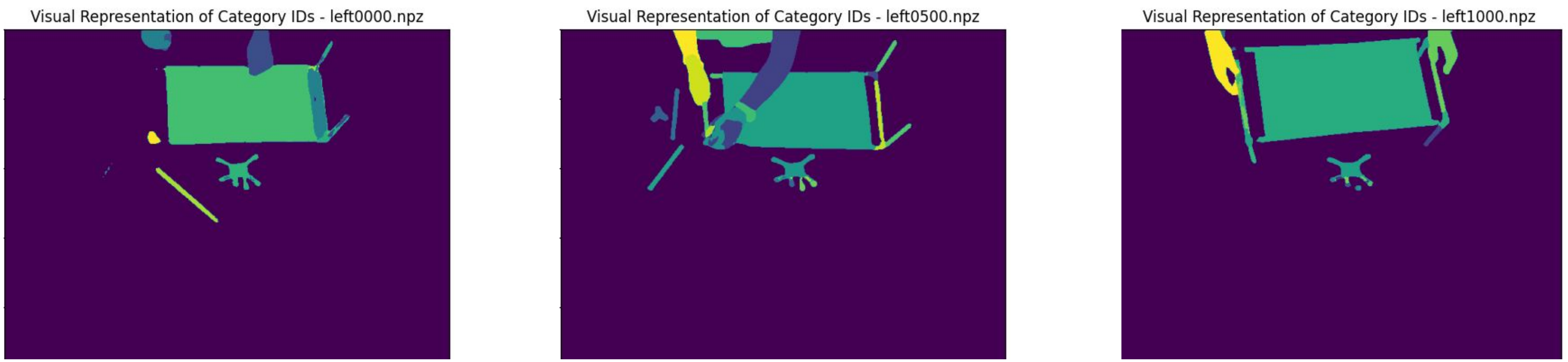


Fig.15: Segmented mask

Visualization - Component Tracking

1) Using the segmentation masks to segment different objects

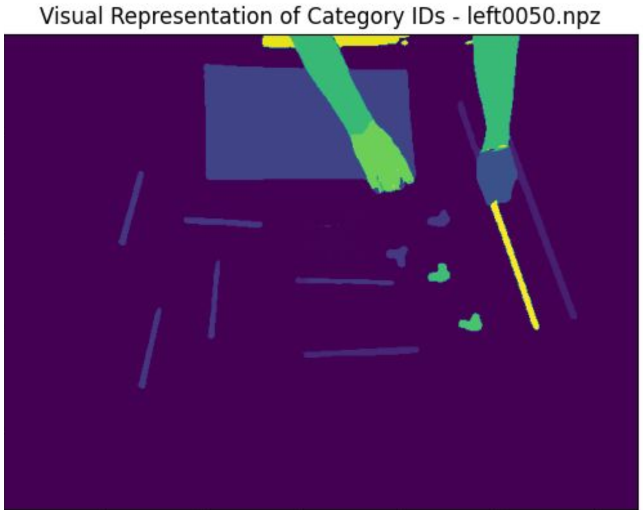


Fig.16: Segmentation mask for frame 50

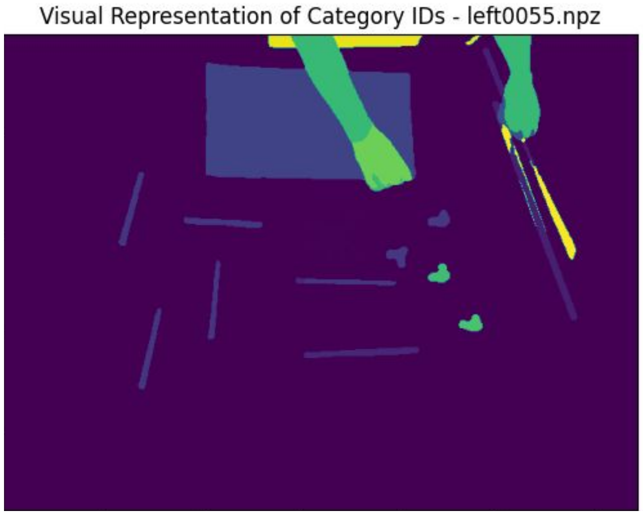


Fig.17: Segmentation mask for frame 55

Visualization - Component Tracking

Step1) Using the segmentation masks to segment different objects

Filtering out the objects

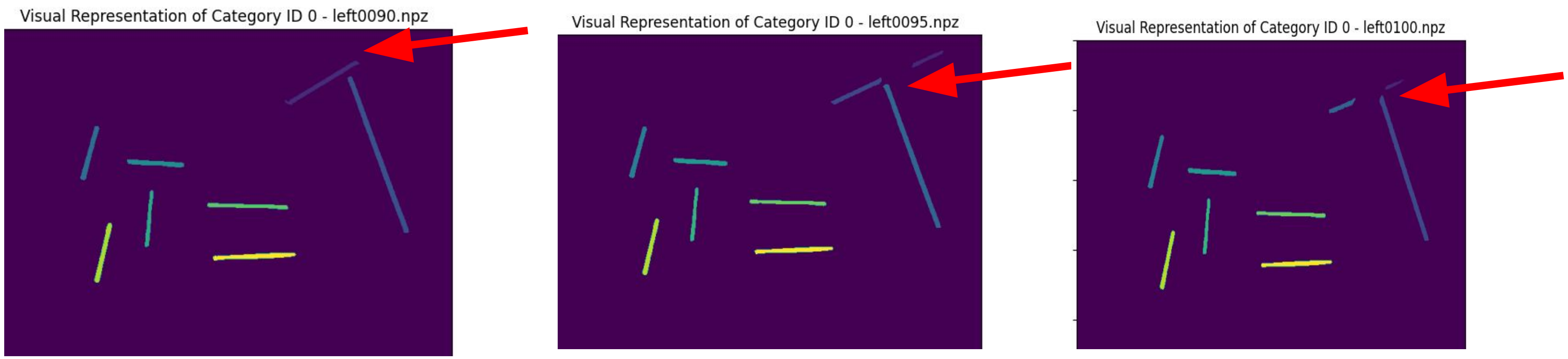


Fig.18: The hand is masking the object and making it discrete

Visualization - Component Tracking

Step2) Using CoTracker to sample the components

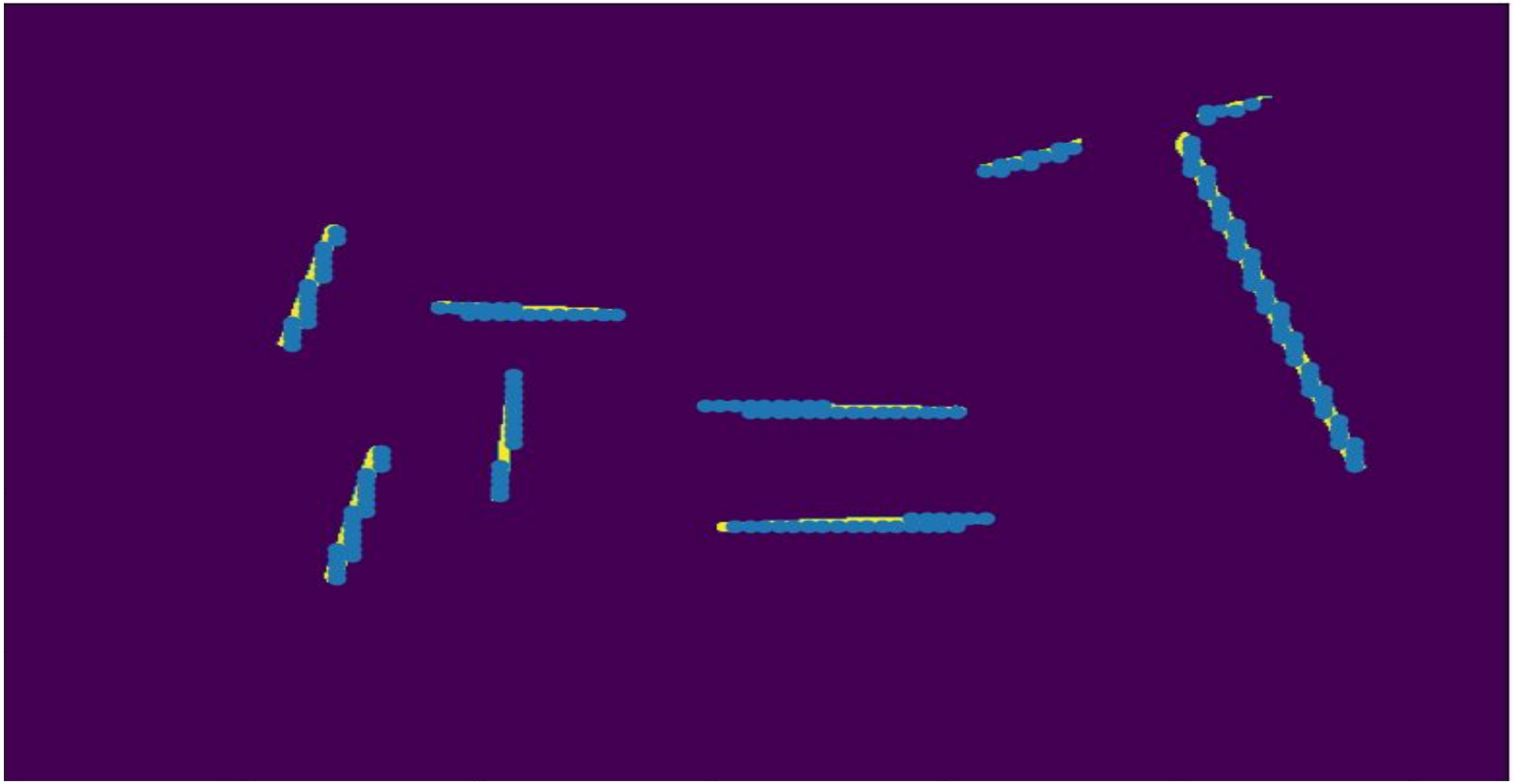


Fig.19: Image points

Visualization - Component Tracking

Step3) Using CoTracker to track the sampled points locations in each sequence (getting the image points)

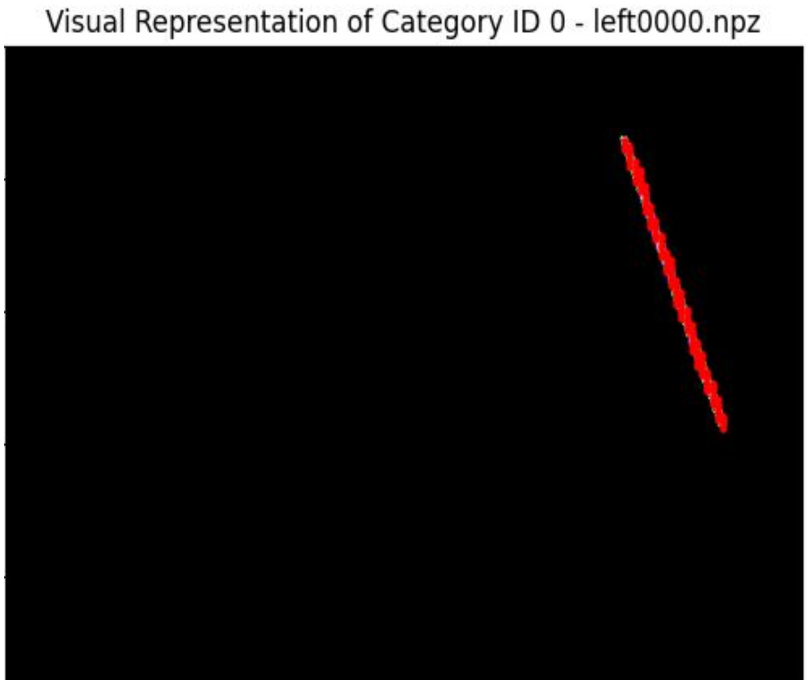


Fig.20: Most points have vanished in the frame 95

Visualization - Component Tracking

Challenges of step 3:

1) *Calibratecamera* function fails to return the camera matrix, distortion coefficients, rotation and translation vectors

Reasons:

1) Error due to numerical instability
2) Insufficient input data
3) CoTracker was not able to extract enough features
   a) Not having a special texture (Like the chessboard)
   b) Object sizes were too small

References

[1]Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, “ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks.” Available: https://arxiv.org/pdf/1910.03151

[2]H. Cheng, Seoung, W. Oh, B. Price, A. Schwing, and J.-Y. Lee, “Tracking Anything with Decoupled Video Segmentation.” Accessed: Jul. 17, 2024. [Online]. Available: https://arxiv.org/pdf/2309.03903

[3]N. Karaev *et al.*, “CoTracker: It is Better to Track Together.” Accessed: Jul. 17, 2024. [Online]. Available: https://arxiv.org/pdf/2307.07635

Thank you!