Eye-Track-ML: A Machine Learning Pipeline for Automated Frame-by-Frame Coding of Eye-Tracking Videos

Mischa Gushiken, Yuexin Li, Jean Ee Tang, Peter Gordon

Language and Cognitive Neuroscience Lab, Teachers College, Columbia University
mkg2145@tc.columbia.edu, yl4964@columbia.edu

Background

- Eye-tracking is an important tool for researchers to understand cognition and behavior. It allows researchers to measure and analyze the direction of a person's gaze providing insight into attention, perception, and decision-making (Bremberg & Cederin, 2023).
- Manual annotation of eye-tracking data is labor-intensive and prone to inter-coder variability, limiting reproducibility (Alinaghi et al., 2024).
- The widespread availability and low cost of machine learning image classification, recognition, and segmentation models make automated annotation a practical and scalable.
- **Motivation for creating pipeline**: Eye-Tracking Study on Infant Event Representation
  - In our investigation of infant event representations, we work with 72 videos (more than 6 hours and over 600,000 frames) of eye-tracking data. The effort required for manual frame-by-frame coding signaled a need for an automated annotation pipeline.

Research Question: How can machine learning models be combined in a pipeline to automate the annotation of fixation points in eye-tracking videos with minimal human intervention?

Literature Review

Current Eye-tracking Annotation Methods

- Eye-tracking annotation methods range from labor-intensive manual coding to more automated approaches. Conventional computer vision techniques—such as detection, clustering, and thumbnail extraction, have given way to approaches using **machine learning (ML)**.
- Recent **ML approaches** have automated fixation point annotation using pre-trained models, such as MYFix's implementation of YOLOv8 and Mask2Former for annotating complex outdoor urban environments (Alinaghi, Hollendonner, Giannopoulos, 2024) and Deep-SAGA's use of Mask R-CNN for automatic gaze annotation with high agreement with manual coders (Deane, Toth, Yeo, 2022).

Computer Vision Components

- **Image classification** models analyze entire images to assign content labels. **YOLO** (You Only Look Once) can perform classification inference. **Object recognition** identifies and localizes multiple objects within frames, with **YOLO** being a popular example and **YOLOv11** being its latest version (Ultralytics, 2024). **Segmentation** models such as **SAM** (Segment Anything Model) create precise pixel-level outlines of objects, with **SAM2.1** being the latest version (Ultralytics, 2024).

Methodology

Overview and Motivation

- Our methodology emerged from a need to code $\sim$ 600,000 frames of eye-tracking data that required identifying precisely what participants were looking at in each frame. Manually coding these data with established coding tools like ELAN would have been prohibitively time-consuming.
- Drawing on our experience with computer vision models, we developed an automated eye-tracking video coding pipeline.

Model Selection and Implementation

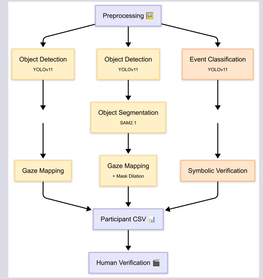
- We explored a variety of approaches in creating our pipeline before settling on our current architecture, which uses an image classification model (YOLOv11), an object detection model (YOLOv11), and an object segmentation model (SAM2.1).
- We trained YOLOv11 and SAM2.1 models—the latest model versions—on 600 diverse frames, with additional training on challenging scenes like hugging where multiple objects appear in close proximity.
- We use YOLOv11 for object detection (providing rectangular bounding boxes that offer an inherent "aura" around objects) and SAM2.1 for contoured object segmentation (using YOLO's detected center points as prompts); we also implement mask dilation post-processing on results from the SAM2.1 model to create an optimal 10px "aura effect" that improves fixation assignment accuracy.

Pipeline Architecture: ML Framework

Pipeline Architecture

Our system processes participant videos through five steps:

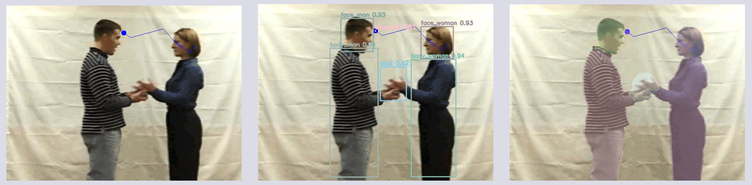
- **First**: Break participant videos into individual frames.
- **Second**: Run image classification inference (YOLOv11) to identify events (hug-with-toy, hug-w/o-toy etc.) in each frame.
- **Third**: For object detection, we have two options: YOLO-only provides rectangular bounding boxes, while YOLO+SAM delivers segmentation masks with object contours.
- **Fourth**: Gaze mapping applies rules to determine what participants are looking at when the gaze indicator (blue dot) occludes objects, grounding abstract gaze coordinates to semantic entities in the scene.
- **Fifth**: Consolidate results into participant CSV datasheets.



Finally, confirm accuracy with human verification using a custom video overlay that displays pipeline results directly on each frame, allowing reviewers to validate and correct the CSV outputs as needed.

Processing Examples

Original | **YOLOv11** | **YOLOv11 + SAM2.1**



(a) Frame before inference; original frame. *(b) Frame after inference; bouding boxes.* *(c) Frame after inference; segmentation masks*

Experimental Validation

We conduct 4 experiments to evaluate distinct machine learning techniques and 1 experiment to evaluate to evaluate symbolic system verification. We compare machine output results to human-verified data.

Experiment 1: YOLO Bounding Box Evaluation

- YOLOv11 object identification performance with bounding boxes versus human-verified data.

Experiment 2: SAM Segmentation Evaluation

- SAM2.1 object segmentation performance with and without mask dilation versus human-verified data.

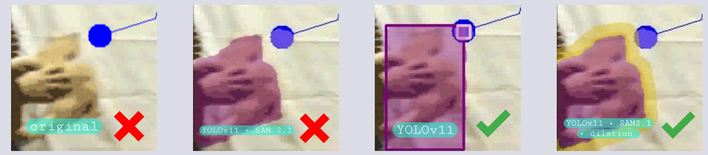
Experiment 3: Classification Evaluation

- YOLOv11 image classification of event type performance versus human-verified data.

Experiment 4: Symbolic Verification Evaluation

- Symbolic verification of classification outputs performance versus human-verified data.

Results



Experiment 1: YOLO-Only Object Detection

- **88.88% overall accuracy** against human-verified data.

Experiment 2: SAM+YOLO Dilation Comparison

- Base model with **no dilation achieved 93.57% accuracy**, demonstrating strong performance.
- **10px dilation optimal at 94.24%** overall accuracy, providing the best results.
- Performance consistently declined as dilation increased beyond 10px.

Experiments 3 & 4: Symbolic Verification for Event Classification

- **99.18% accuracy**. Exceptional reliability in event classification.
- **100% accuracy** after symbolic system verification.

Overall Pipeline Effectiveness

- The full pipeline achieves near-human level accuracy with minimal verification needed.
- Combined SAM+YOLO (with 10px dilation) significantly outperformed the YOLO-only approach by 5.36%.

Discussion

- The SAM+YOLO approach with 10px mask dilation achieved the highest accuracy at **94.24%**. We attribute this to SAM's ability to precisely capture object contours, which we enhanced through mask post-processing to create the "aura effect" essential for our specific study. We fine-tuned our models to address low-resolution videos, grainy objects, and tracking "invisible" objects, which requires defining bounding boxes for empty spaces.
- Verification of ML outputs with symbolic system verification proved highly effective, achieving **100%** accuracy in event classification. While our implementation focused on a relatively straightforward verification case, our success in combining statistical outputs with logical rules demonstrates potential for further automating eye-tracking data collection.
- Despite the high accuracy of our automated approach, human verification remains essential for detecting subtle patterns, saccades, and judging edge cases. Nevertheless, our system establishes a strong baseline for consistency, requiring human verifiers to correct only $\sim$ 6% of data points.

References

1. Alinaghi, N., Hollendonner, S., Giannopoulos, I. (2024). MYFix: Automated fixation annotation of eye-tracking videos. Sensors, 24, 2666. https://doi.org/10.3390/s24092666
2. Bremberg, U., Cederin, L. (2023). Automatic object detection and tracking for eye-tracking analysis [Master's thesis, Uppsala University]. Uppsala University.
3. Deane, O., Toth, E., Yeo, S.-H. (2022). Deep-SAGA: A deep-learning-based system for automatic gaze annotation from eye-tracking data. Behavior Research Methods, 55, 1372–1391. https://doi.org/10.3758/s13428-022-01833-4
4. Ultralytics. (2024). SAM 2 (segment anything model 2) [Documentation]. Ultralytics YOLO Docs. https://docs.ultralytics.com/models/sam-2/
5. Ultralytics. (2024). YOLO11 [Documentation]. Ultralytics YOLO Docs. https://docs.ultralytics.com/models/yolo11/