# Exploring Aspects of Social Interaction using Machine Learning

Nele-Pauline Suffo[1], Pierre-Etienne Martin[2], Anam Zahra[2], Daniel Haun[2], & Manuel Bohn[1, 2]

[1] Institute of Psychology in Education, Leuphana University Lüneburg

[2] Max Planck Institute for Evolutionary Anthropology

## Author Note

The authors made the following contributions. Nele-Pauline Suffo: Conceptualization, Writing - Original Draft Preparation, Writing - Review & Editing; Manuel Bohn: Writing - Review & Editing, Supervision.

Correspondence concerning this article should be addressed to Nele-Pauline Suffo, Universitätsallee 1, 21335 Lüneburg. E-mail: nele.suffo@leuphana.de

Abstract

tbd

Exploring Aspects of Social Interaction using Machine Learning

## Introduction

## Methodology

The Quantex dataset includes

### Dataset Description

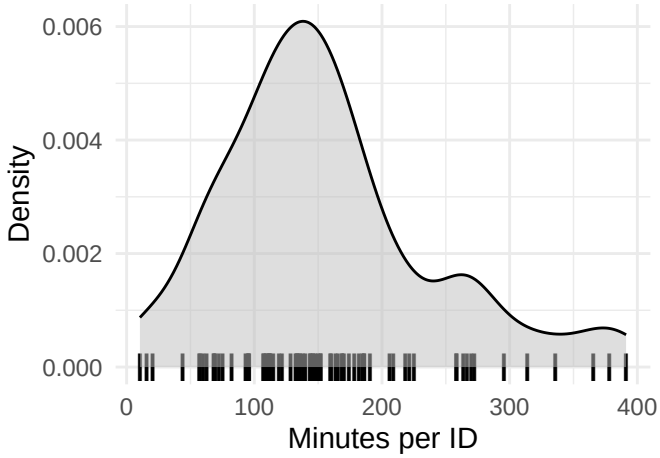


*Figure 1*. Video recording duration (in minutes) per Child in the Quantex Dataset.

#### Statistics.

**Annotation Strategy.** The dataset annotations cover four key elements: persons, faces, gaze direction, objects the child interacts with. Gaze information identifies whether a detected person's gaze is directed toward the child or not. For every detected person (or reflection of a person, such as in a mirror) and face, additional attributes like age and gender are collected. Objects are categorized into six distinct groups: book, screen, animal, food, toy, and kitchenware, with an additional category for other objects. The dataset focus is on detecting and labeling instances of (social) interaction and engagement through these key categories. The annotation strategy is displayed in Figure 2.

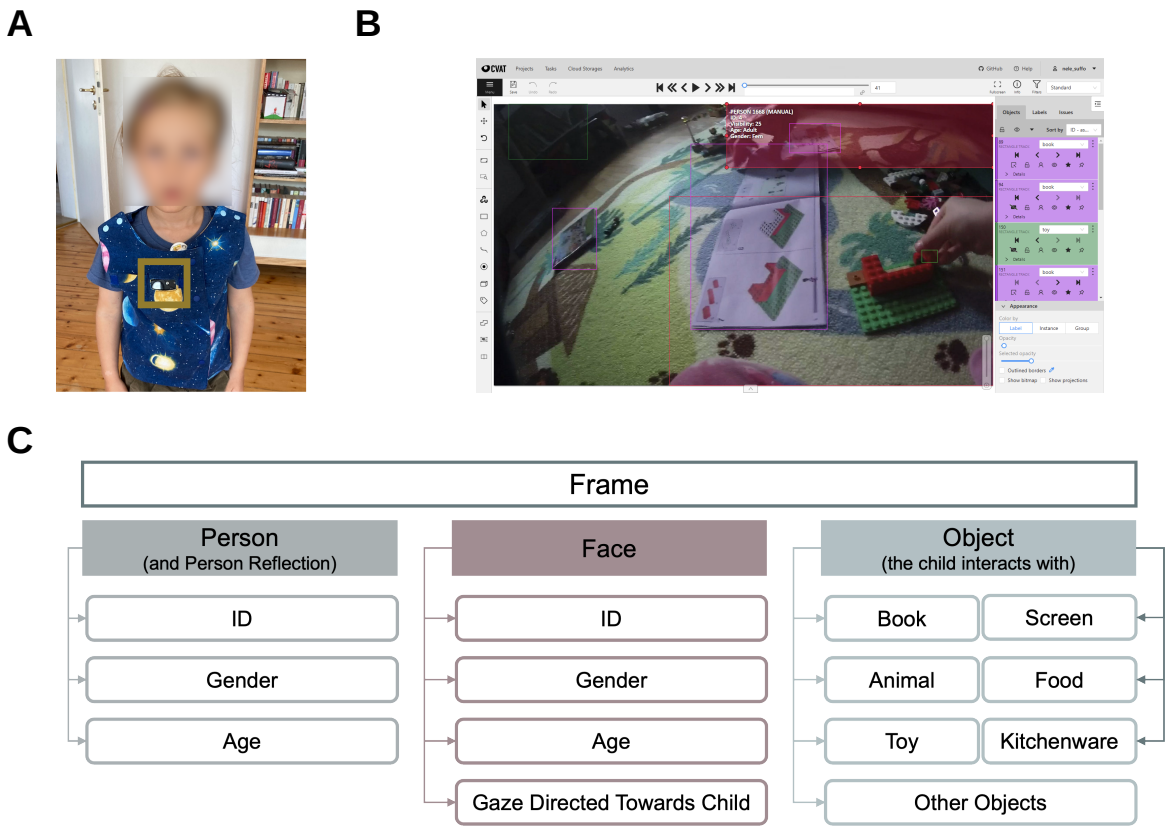


*Figure 2*. **A** – Vest with the embedded camera worn by the children, **B** – CVAT platform utilized for video annotation, **C** – Annotation Strategy in the Quantex dataset.

## Data Collection

This study collected egocentric video recordings from 76 children, aged 3 to 5 years, over a span of 73 months. Participating families lived in a mid-sized city in Germany. To capture the children's everyday experiences, a wearable vest equipped with a camera was used, as shown in figure 2. The camera, a *PatrolEyes WiFi HD Infrared Police Body Camera*, provided high-definition video (1920x1080p at 30 fps) with a 140-degree wide-angle lens and also recorded audio. Children were free to move around and engage in their usual activities at home without any interference or instructions given to their parents.

Table 1

*Evaluation metrics for the YOLO11 face detection model trained on the Quantex dataset.*

| Dataset | Precision | Recall | F1-Score |
|---|---|---|---|
| Quantex | 0.90 | 0.83 | 0.86 |

## Data Preprocessing

For the video data, the annotation strategy required persons, faces, and objects to be labeled even when only partially visible, as long as key features such as facial landmarks (e.g., nose, eye, or mouth) or parts of a person or object were clearly visible. Frames that were too blurry due to movement were marked as "noise" and excluded from further analysis. Additionally, frames where the child was not wearing the camera, as well as any scenes containing nudity, were also labele d as noise and removed from the dataset. To prepare the video data for analysis, one frame per second was annotated, corresponding to every 30th frame in the video. Similarly, every 30th raw frame was extracted from the annotated video files. No preprocessing was applied to the audio data, which was used in its raw form for analysis.

## Automated Analysis Pipeline

### Person Detection.

### Face Detection.

### Gaze Classification.

### Voice Detection and Classification.

### Feature Extraction

## Results

### Presence of Aspects of Social Interaction

#### Presence of a Person.

#### Presence of a Face.

#### Presence of Gaze Directed at the Child.

#### Presence of Language.

### Co-occurrence of Aspects of Social Interaction

## General Discussion

## References

## Supplementary Material

### Yolo Face Detection Model Training Details

Face detection is perfored utilizing Ultralytics Yolo11 (Jocher & Qiu, 2024) We employed a YOLO11 model pretrained for face detection (Codd, 2024), which was fine-tuned on our dataset to adapt it to the unique characteristics of our egocentric dataset, captured using chest-mounted cameras. While we initially experimented with the MTCNN model, its performance on our dataset proved insufficient. Consequently, we chose YOLO due to its streamlined training process and fewer requirements for data preparation. The 100 annotated videos were divided into 70% for training, 10% for validation, and 20% for testing. This split corresponded to 51 with 72687 frames for training, 6 videos with 7720 frames for validation, and 7 videos with 9272 frames for testing.

Model training was conducted using the Ultralytics framework (Jocher, Jing, & Chaurasia, 2023) on a Linux server equipped with 48 cores and 187 GB of RAM. The training process utilized YOLO's built-in data augmentation, a batch size of 16, a cosine annealing learning rate scheduler, and early stopping after 10 epochs without improvement, with a maximum of 200 epochs. Training concluded after 86 epochs, achieving a precision of 0.90 and a recall of 0.83, resulting in an $F_1$-score of 0.86 on the testing set. This indicates strong performance in correctly identifying most faces while minimizing errors, although some challenges remain. These performance metrics, summarized in Table 1, underscore the model's ability to reliably detect faces, with further details and evaluation available in the supplementary materials.

The model performed well in detecting faces, particularly when fully visible from the front, but few challenges remain in more dynamic scenarios. For example, faces that are partially visible, rotated, or seen from the side often resulted in detection errors. Furthermore, false negatives were more common when faces were occluded by the child's body, blurred due to movement, or situated in the background. While background faces are

less relevant, as they are unlikely to be part of an interaction with the child, missed detections due to occlusions or motion blur present a greater challenge. In these cases, we rely on adjacent frames to provide clearer views for more accurate classification. These difficulties underscore the challenges of working with egocentric video data, where dynamic movement and varying perspectives, typical of chest-mounted camera recordings, introduce additional complexity.

Accurate face detection remains a crucial step in our automated analysis pipeline, as it serves as the foundation for subsequent gaze classification. Identifying the presence and position of faces ensures that gaze direction can be reliably analyzed, allowing us to determine when and how individuals engage with the child.

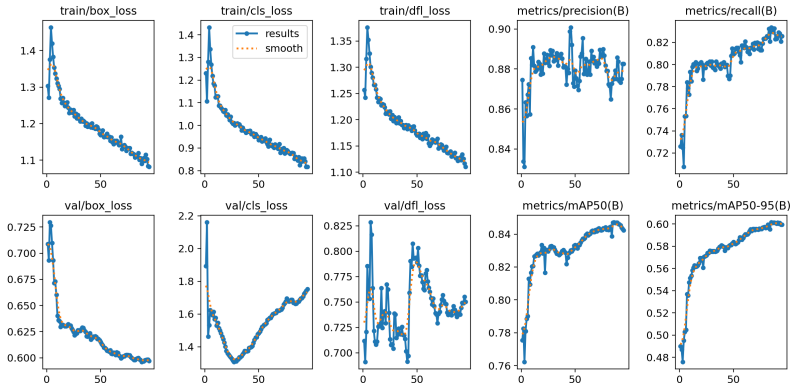


*Figure 3*. Training and Validation Loss Curves for the YOLO11 face detection model.

**Model Training Details.**

- Yolo11 uses a loss function consisting of three components: box loss, class loss, and distribution focal loss (DFL) as displayed in 3 (Codd, 2024).
- Box Loss: difference between expected bounding boxes and ground truth boxes.
- Class Loss (Cls Loss): Measures the model's ability to accurately identify detected items which is less relevant in our case as we are only interested in one class: faces

- Distribution Focal Loss (DFL) improves the model's ability to identify difficult-to-detect objects or classes
- we can see that during training all three loss components decrease over time, indicating that the model is learning and improving its performance.
- more precisely, decrease in box loss indicates that the model is becoming more precise at localizing items within images, classification loss converges quickly showing that the model is able to identify faces accurately, and the distribution focal loss decreases over time, indicating that the model is learning to focus on difficult-to-detect objects or classes, improving its overall performance.

**Model Evaluation Metrics.** Jocher (2024)

- when looking at confusion matrix: you can see that the majority of images are correctly identified as face. 0.86% of all faces are correctly identified by the model, resulting in a total of 1000 faces detected, while r num_faces_missed‘ weren't detected.
- looking into those false negative cases (meaning model predicts no face but label says face): we can see that the majority of those cases are due to three reasons: faces in the background, blurry faces or faces that are occluded by the child's body.
- (**ref?**)(fig:face-metrics) provides examples of correctly classified face images (true positives) and incorrectly classified face images (false negatives). Regarding images in the background were less concerned as they are unlikely to be part of an interaction with the child. However, the other two cases are more critical as they are likely to be part of an interaction with the child.
- here we have to rely on the surrounding frames that hopefully contain a clearer view of the face to make a more accurate classification.
- a small amount of images are classified as faces while no face is present in the image

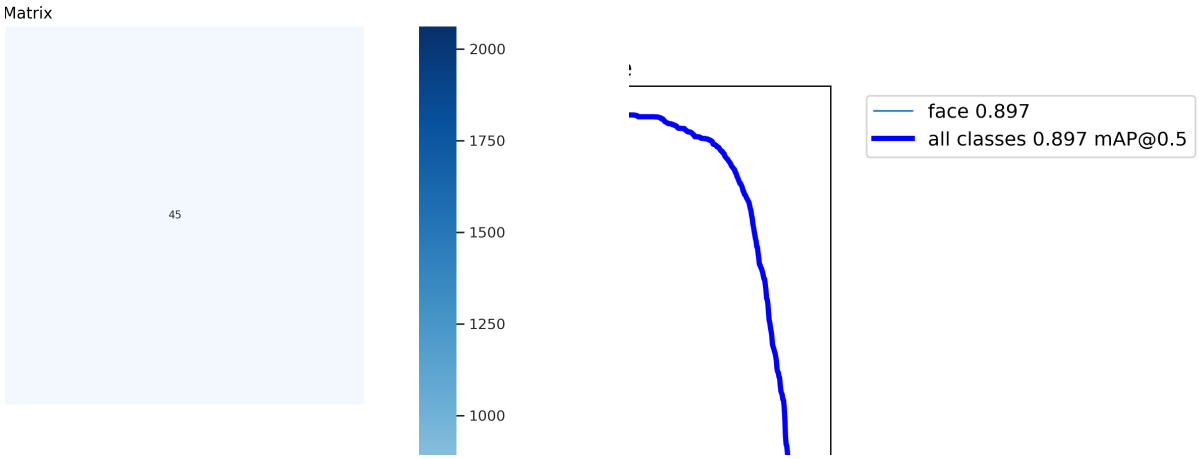


*Figure 4*. Evaluation metrics for the YOLO11 face detection model trained on the Quantex dataset.

(false positive). from the confusion matrix you can derive that the model only classifies 0.86 images, corresponding to

This is due to the model's tendency to detect faces in objects that resemble faces, such as toys or drawings.

- when looking at the precision-recall curve: we can see that the model performs well in terms of precision and recall, with a high precision and recall value, indicating that the model is able to detect faces accurately and with high confidence.
- the curve stays close to the top left corner, indicating that the model is able to detect faces with high precision and recall, which is crucial for our analysis pipeline.

4. Model Hyperparameter Settings and Tuning • Hyperparameter Search Results: If you did any hyperparameter tuning (e.g., batch size, learning rate), you could include a plot showing how different values impacted performance. • Plot type: A heatmap or line plot showing the effect of different hyperparameter combinations on a given metric (e.g., F1-score).

5. Training Time and Hardware Utilization • Training Time Breakdown: Include the total time taken for training and the hardware specifications (e.g., time per epoch, total training time). • Plot type: A bar graph showing training time per epoch or the total time required for training. • Resource Utilization (optional): If you tracked resource usage during training (e.g., GPU/CPU utilization, memory usage), you can provide plots to show how efficiently the hardware was used.

Codd, A. (2024). *YOLOv11n-face-detection*. Retrieved from https://huggingface.co/AdamCodd/YOLOv11n-face-detection

Jocher, G. (2024). *Ultralytics Yolo11 Performance Metrics*. Retrieved from https://docs.ultralytics.com/guides/yolo-performance-metrics/?utm_source=chatgpt.com

Jocher, G., Jing, Q., & Chaurasia, A. (2023). *Ultralytics YOLO*. Retrieved from https://github.com/ultralytics/ultralytics

Jocher, G., & Qiu, J. (2024). *Ultralytics YOLO11*. Retrieved from https://github.com/ultralytics/ultralytics

# Appendix