Zoom Behavior Insight

Presented by:
Ofir Duek , Aviv Meir , Rotem Aloni

Project Overview

We build a model that maps student images from Zoom classes to a predefined discrete behavior vector describing observable learning-related attributes such as gaze direction, environment type, presence of headphones, and objects held in hand .

Problem & Use Case

The Problem :
In large Zoom based classes, instructors lack visibility into individual student behavior during the lesson.

Impact:
It is difficult to assess student engagement and identify potential distractions based solely on the Zoom interface.

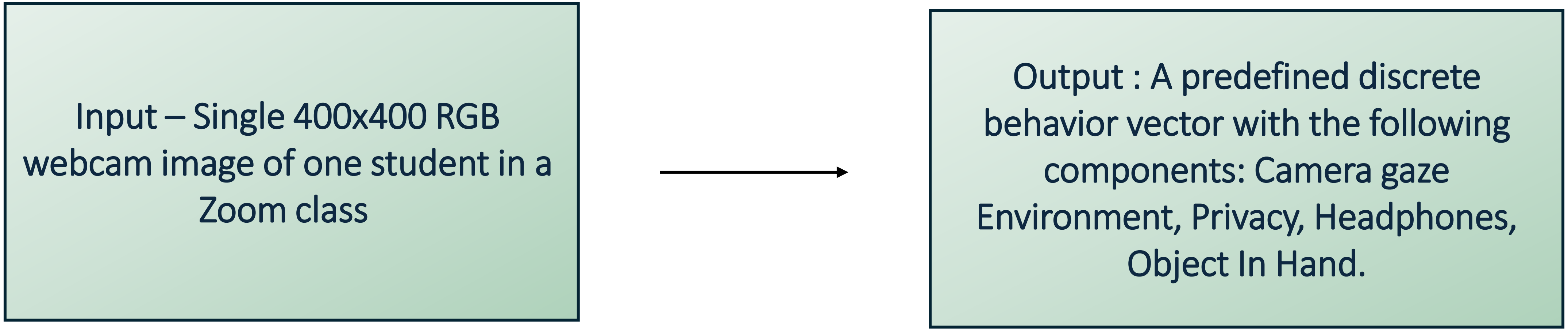
Our Approach:
Automatically extract a structured behavior representation for each student from a single Zoom frame.

Example:
For each student, the system outputs a discrete behavior vector:

- Gaze : Camera
- Environment : Indoor
- Privacy : Private
- Headphones : WithoutHeadphones
- ObjectInHand : Phone

ML Task & Output Definition

Task : Multi-task image classification for extracting discrete behavioral attributes from student images in Zoom frames.



Input – Single 400x400 RGB webcam image of one student in a Zoom class

Output : A predefined discrete behavior vector with the following components: Camera gaze Environment, Privacy, Headphones, Object In Hand.

Example 1: Student looking at the screen/camera, in a quiet room, writing notes.

Example output:

- Gaze: Camera
- Environment: Indoor
- Privacy: Private
- Headphones: Without Headphones
- Object In Hand: Pen

Example 2: Student looking down at a phone, slightly turned away from the screen.

Example output:

- Gaze: Not Camera
- Environment : Outdoor
- Privacy : Public
- Headphones : Without Headphones
- Object In Hand : Phone

Dataset Overview

Data Sources:

- Real webcam-like images of students sitting in front of a laptop.
- Synthetic images generated with Stable Diffusion + inpainting using prompts designed to control the predefined behavior attributes, including gaze direction, environment type, background privacy, presence of headphones, and objects held in hand.

Annotations (Labels):

Each image is annotated with a predefined discrete behavior vector, according to the labeling guidelines.

The behavior vector includes the following attributes:

- Gaze (Camera / Not camera / Eyes closed)
- Environment (Indoor / Outdoor)
- Privacy (Private / Public)
- Headphones (With headphones / Without headphones / Unknown)
- Object in hand (Phone / Pen / Cup / Other / None / Unknown)

Annotations are stored per image in a structured CSV file, Labels are stored in a CSV file together
with the image filename.

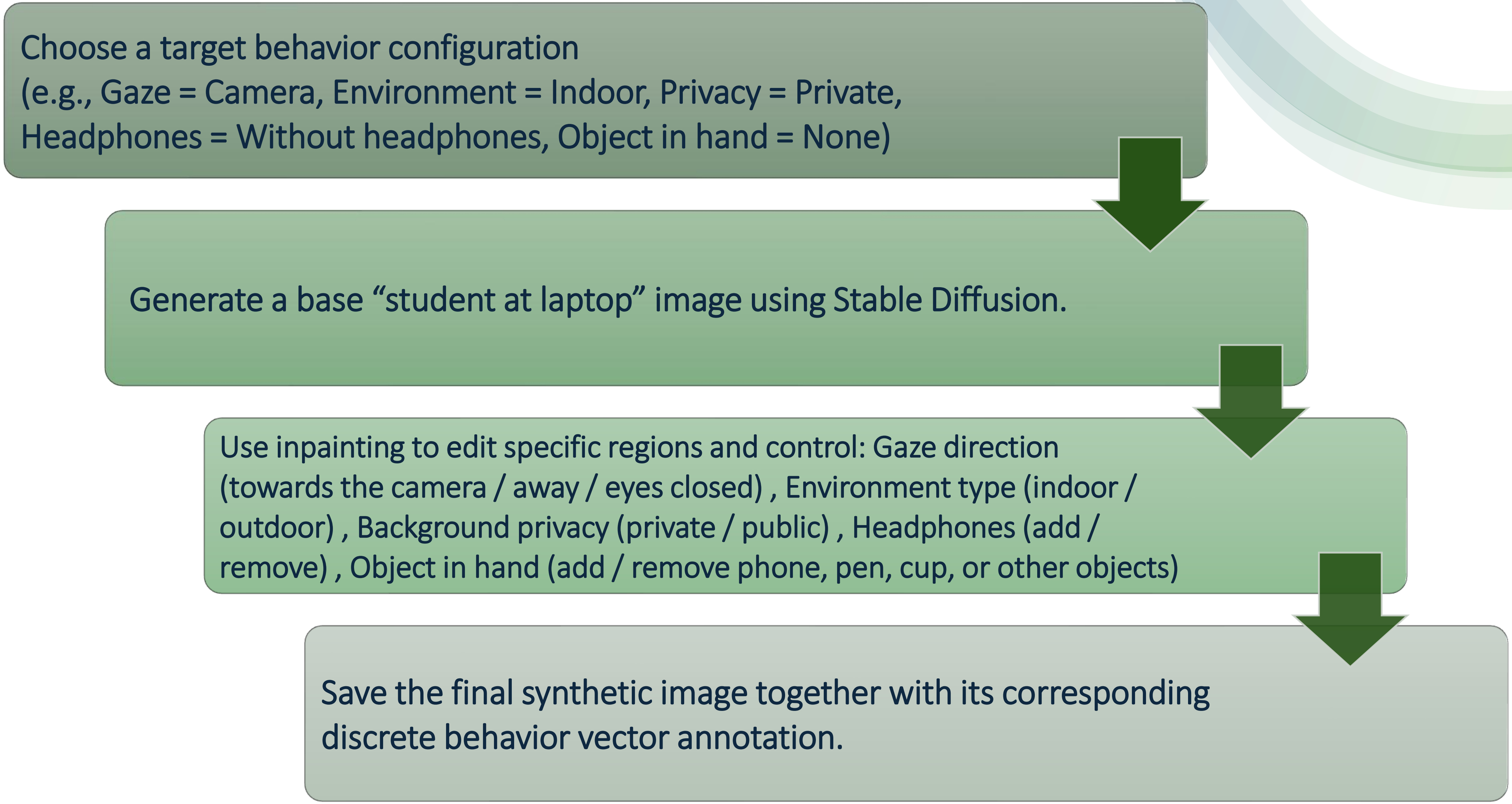
Data Split

- Train 80%, validation 10%, test 10%.
- Test set will use only real images to measure performance on real data.

Size & Format

- Target: ~1000-2000 images in total.
- RGB, 400x400 JPG/PNG files.

Data Synthetic Generation Pipeline



Choose a target behavior configuration
(e.g., Gaze = Camera, Environment = Indoor, Privacy = Private,
Headphones = Without headphones, Object in hand = None)

Generate a base “student at laptop” image using Stable Diffusion.

Use inpainting to edit specific regions and control: Gaze direction (towards the camera / away / eyes closed) , Environment type (indoor / outdoor) , Background privacy (private / public) , Headphones (add / remove) , Object in hand (add / remove phone, pen, cup, or other objects)

Save the final synthetic image together with its corresponding discrete behavior vector annotation.

Behavior Control & Prompt Examples

What we control in the synthetic images:

- Gaze direction – Camera / Not camera / Eyes closed
- Environment type – Indoor / Outdoor
- Background privacy – Private / Public
- Headphones – With headphones / Without headphones
- Object in hand – Phone / Pen / Cup / Other / None

Examples:

1. Prompt example - Student sitting at a desk, looking at the laptop screen, indoor study room, no other people in the background, no headphones, empty hands.
2. Prompt example – Student sitting in a busy café, looking down at a smartphone, wearing headphones, people visible in the background.

Model & Training Setup

We use a pretrained CNN and adapt it to predict a predefined discrete behavior vector for each student image.

Model:

- Pretrained CNN backbone (e.g., ResNet-18)
- Replace the final layer with multiple output heads, one per behavior attribute : Gaze, Environment, Privacy, Headphones, Object In Hand.
- Each head predicts a discrete class (softmax) for its attribute.

Training setup:

- Input preprocessing
  - Resize each image to 400x400 RGB.
  - Normalize pixel values.
  - Simple data augmentation (random horizontal flip, small brightness/contrast changes).
- Loss & optimizer
  - Loss: Cross-entropy loss applied separately to each behavior attribute.
  - Optimizer - Adam with a small learning rate.
- Training modes
  - Baseline: Freeze the CNN backbone and train only the output heads.
  - Fine-tuned model: Unfreeze the last CNN layers and train them jointly with the output heads.

Experiments & Metrics

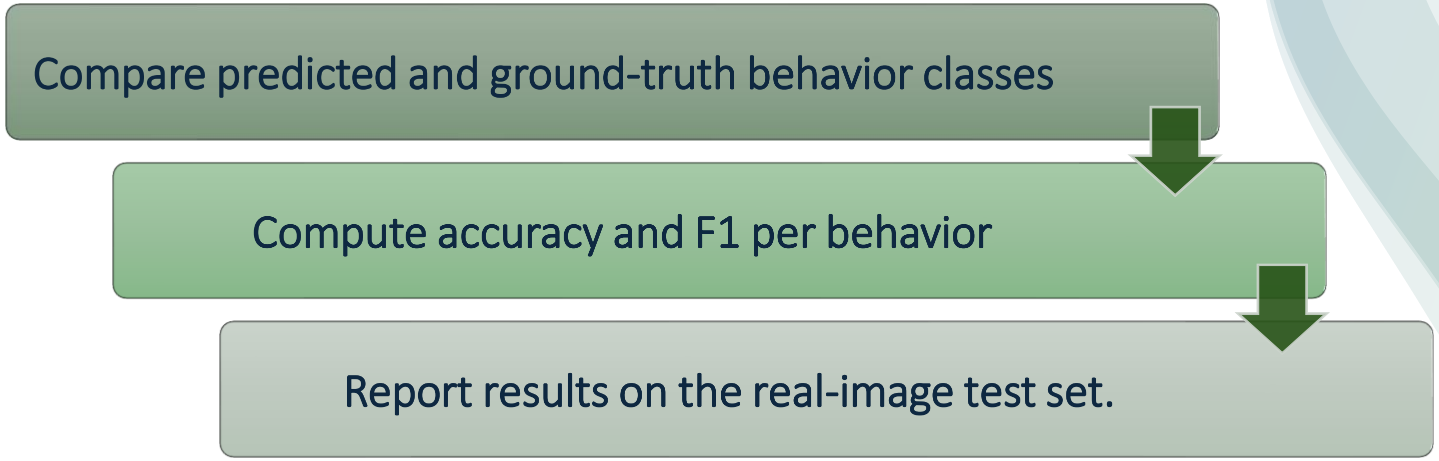
We define the evaluation metrics and the experimental setup used to assess the model.

Metrics :

We evaluate the model using

- Accuracy per behavior attribute (Gaze, Environment, Privacy, Headphones, Object In Hand)
- Macro F1-score computed separately for each attribute

Evaluation procedure



Compare predicted and ground-truth behavior classes

Compute accuracy and F1 per behavior

Report results on the real-image test set.

Planned experiments

1. Baseline vs. fine-tuned CNN:
   Same train/validation data (real + synthetic)
   - Model 1 : backbone frozen, train only the output heads.
   - Model 2 : fine-tune the last CNN layers together with the output heads.
2. Real-only vs. real + synthetic data: Same fine-tuned architecture
   - Train on real images only.
   - Train on real + synthetic images.
   - Evaluate both models on the same real-image test set.

Novelty & Contribution

- Behavior-based representation of Zoom webcam images:
  Each student image is mapped to a predefined discrete behavior vector, rather than a single label.

- Synthetic Zoom-like dataset with controllable behaviors:
  We use Stable Diffusion and inpainting to generate and edit student images , while explicitly controlling the predefined behavior attributes.

- Systematic evaluation of model and data setups:
  We compare baseline vs fine-tuned CNNs and real-only vs real + synthetic training to evaluate performance on real Zoom webcam images.

Risks, Limitations & Next Steps

Risks & limitations:

- Ambiguous images where behavior attributes are hard to label (e.g., unclear gaze, partially visible objects)
- Synthetic images may not perfectly match real Zoom webcam frames
- Limited size and diversity of the real-image dataset

Next steps:

- Finalize data collection and labeling guidelines for the predefined behavior attributes.
- Train the baseline and fine-tuned CNN models.
- Run the experiments and analyze classification performance per behavior attribute on the real-image test set.