Integrating Faze Estimator into the project and training it on the GazeCapture dataset involves following steps:

Preprocessing the Dataset:

This involves preparing the GazeCapture dataset by loading images, extracting relevant information such as head poses and gaze directions, and potentially resizing or normalizing the images.

Detecting Faces and Eyes:

Face and eye detection can be performed using techniques like Haar cascades or deep learning-based methods such as convolutional neural networks (CNNs). OpenCV provides pre-trained models for face and eye detection.

Image Normalization:

Normalizing the images ensures consistent input to the model, which can include resizing images to a standard size, converting to grayscale, and applying techniques like histogram equalization for improved contrast.

Converting Images to Tensors:

To input images into a deep-learning model, they need to be converted into tensors. Libraries like TensorFlow or PyTorch can be used for this purpose.

Training the Faze Estimator:

The Faze Estimator model can be built using deep learning techniques such as CNNs or recurrent neural networks (RNNs). The model architecture should be designed to take preprocessed images as input and predict gaze directions as output.

Evaluation and Fine-Tuning:

After training, the model will be evaluated on a separate test set to assess its performance. Fine-tuning can be performed by adjusting hyperparameters or modifying the model architecture based on evaluation results.