Institute of Parallel and Distributed Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit Nr. 3570210

Contrastive Representation Learning for Eye Contact Detection

Zhenhao Xu

Course of Study:	Autonome System
Examiner:	Prof. Dr. Andreas Bulling
Supervisor:	Dr. Mihai Bâce
Commenced:	15. June 2023
Completed:	15. December 2023
CR-Classification:	I.7.2

Abstract

While extensive research has been dedicated to gaze estimation, featuring numerous methods and datasets, eye contact detection, despite receiving comparatively less attention and marked by a scarcity of datasets, still holds significant practical applications. For instance, in remote learning scenarios, eye contact detection can be employed to ascertain whether students are focusing their attention on the screen. This technology can be instrumental in enhancing virtual engagement and educational efficacy. Moreover, the challenge in generalizing between the datasets of gaze estimation and eye contact detection, mainly due to their differing labeling approaches, poses a significant challenge. These challenges, particularly the scarcity of dedicated datasets and the difficulty in direct application of gaze estimation methods to eye contact detection, necessitate a novel approach.

In response to these issues, this thesis introduces a novel approach to model construction for eye contact detection, employing an unsupervised contrastive learning method. This method was chosen for its ability to utilize large amounts of unlabeled data from gaze estimation datasets, particularly advantageous given the scarcity of dedicated eye contact detection datasets. In our study, we employed the SimCLR contrastive learning model, optimized specifically for eye contact detection. This optimization led to a significant improvement in the Matthews Correlation Coefficient (MCC) for eye contact detection, elevating it from 0.46, as achieved by Zhang et al.'s method, to 0.63 with our approach. Notably, our method achieves this enhanced performance without the need for datasets manually labeled with gaze direction or eye contact labels. This marks the pioneering application of contrastive learning to the task of eye contact detection, showcasing its efficacy in improving key performance metrics.

Additionally, in the fine-tuning process of our contrastive learning model, while there was still a requirement for a small dataset labeled with eye contact detection, we sought to completely eliminate the dependency on manually annotated eye contact labels. To achieve this, we utilized state-of-the-art gaze estimation models, not as the primary method, but as an auxiliary tool to automatically generate pseudo-labels for eye contact detection. This strategy effectively leverages the outputs of the gaze estimation models to produce reliable pseudo-labels, allowing our eye contact detection model to operate independently of manual labeling.

Kurzfassung

Während der Blickschätzung, die zahlreiche Methoden und Datensätze umfasst, umfangreiche Forschung gewidmet wurde, erhält die Erkennung von Augenkontakt vergleichsweise weniger Aufmerksamkeit und leidet unter einem Mangel an spezifischen Datensätzen. Trotzdem hat sie bedeutende praktische Anwendungen. Zum Beispiel kann in Fernlernsituationen die Erkennung von Augenkontakt eingesetzt werden, um zu überprüfen, ob Schüler sich auf den Bildschirm konzentrieren. Diese Technologie kann entscheidend sein, um virtuelles Engagement und Bildungseffektivität zu verbessern. Darüber hinaus stellt die Herausforderung, zwischen den Datensätzen der Blickschätzung und Augenkontakterkennung zu verallgemeinern, ein bedeutendes Problem dar, vor allem wegen ihrer unterschiedlichen Kennzeichnungsmethoden.

Als Reaktion auf diese Probleme führt diese Arbeit einen neuen Ansatz für die Modellerstellung zur Erkennung von Augenkontakt ein, indem sie eine unüberwachte kontrastive Lernmethode verwendet. Diese Methode wurde wegen ihrer Fähigkeit ausgewählt, große Mengen an unbeschrifteten Daten aus Datensätzen zur Blickschätzung zu nutzen, was besonders vorteilhaft ist, angesichts des Mangels an speziellen Datensätzen zur Erkennung von Augenkontakt. In unserer Studie verwendeten wir das SimCLR kontrastive Lernmodell, das speziell für die Erkennung von Augenkontakt optimiert wurde. Diese Optimierung führte zu einer deutlichen Verbesserung des Matthews-Korrelationskoeffizienten (MCC) für die Erkennung von Augenkontakt.

Im Rahmen des Feinabstimmungsprozesses unseres kontrastiven Lernmodells bestand zwar weiterhin die Anforderung an einen kleinen, mit Augenkontakt-Erkennungslabels versehenen Datensatz, doch zielten wir darauf ab, die Abhängigkeit von manuell annotierten Augenkontakt-Labels vollständig zu eliminieren. Dazu nutzten wir hochmoderne Modelle zur Blickschätzung, nicht als primäre Methode, sondern als Hilfsmittel zur automatischen Generierung von Pseudo-Labels für die Augenkontakterkennung. Diese Strategie nutzt effektiv die Ausgaben der Blickschätzungsmodelle, um zuverlässige Pseudo-Labels zu erzeugen, was es unserem Augenkontakterkennungsmodell ermöglicht, unabhängig von manueller Kennzeichnung zu operieren.

Contents

List of Figures

List of Tables

1 Introduction

Mobile devices, such as smartphones, have become ubiquitous in daily life, yet users often exhibit scattered attention. Understanding the distribution of a user's attention is crucial for designing effective human-computer interaction interfaces capable of managing limited attention spans. To quantify a user's visual attention on a mobile device, a straightforward method involves detecting whether the user is looking at it[ZSB17]. This method, referred to as 'eye contact detection', follows the terminology established by Zhang et al., where it specifically relates to maintaining eye contact with a target, such as a mobile phone in this study. More extensive research has been conducted on gaze estimation compared to that on eye contact detection.

With the advancements in deep learning, estimating a user's gaze direction using a mobile device's camera has become increasingly feasible, eliminating the need for specialized eye-tracking equipment. The field of gaze estimation has seen a proliferation of research and the development of various appearance-based methods. These methods are capable of determining the 3D gaze direction by analyzing the overall appearance of the user's face and head pose, typically through a regression task that predicts continuous eye gaze directions.

In contrast, eye contact detection, primarily a binary classification task, aims to ascertain whether the user is looking at their device. While closely related to gaze estimation, the methods and research for eye contact detection are considerably less developed. The direct application of gaze estimation techniques for eye contact detection often leads to significant inaccuracies, as gaze estimation alone may not accurately predict gaze location or infer eye contact [ZSB17]. This disparity suggests the opportunity to leverage existing gaze estimation methods and datasets for the development of more effective models for eye contact detection.

The collection of datasets for eye contact detection and gaze estimation poses significant challenges, primarily due to the reliance on eye trackers which are often confined to lab settings and not widely accessible. Hence, an approach like contrastive learning, which can learn effectively from unlabeled data, becomes crucial. In light of this, we utilize contrastive learning frameworks to construct models specifically for eye contact detection tasks in this paper. Due to the absence of a dedicated dataset and annotations for this task, we leverage datasets from gaze estimation. Ignoring annotations for

3D gaze direction, we introduce an unsupervised training approach using contrastive learning. Following this, fine-tuning for the downstream model is carried out using a small dataset and eye contact labels. This marks the first application of contrastive learning in the context of eye contact detection task.

Contrastive learning, known for its efficacy in extracting meaningful and discriminative representations, plays a pivotal role in both gaze estimation and eye contact detection tasks. By explicitly comparing similar and dissimilar instances during the learning process, this approach enables models to capture underlying patterns, enhancing their ability to generalize and perform effectively across diverse tasks and datasets. In this paper, we primarily focus on applying the principles of contrastive learning to the task of eye contact detection. Our approach not only reduces the dependency on labeled datasets but also significantly enhances the performance of the eye contact detection models.

Building on previous research that optimized contrastive learning for gaze estimation tasks, we explored its applicability and effectiveness for eye contact detection. We found that many optimization techniques for contrastive learning in gaze estimation also proved effective for eye contact detection. This is primarily because these contrastive learning methods can extract features with strong generalizability, suitable for both gaze estimation and eye contact detection tasks, a feat that is challenging to achieve with supervised learning methods. Consequently, our research contributes to the advancement of models in eye contact detection by leveraging the robust feature extraction capabilities of contrastive learning.

Outline

The work is structured as follows.

Chapter 2 – Background: This chapter introduces the concepts relevant to this research. These concepts include neural networks, convolutional operations, support vector machine, multi-layer perception, resnet, contrastive learning.

Chapter 3 – Related Work: This chapter introduces the related Literature on Gaze Estimation and Contrastive Learning

Chapter 4 – Appearance-based Gaze Estimation: This chapter delineates the training pipeline, commencing with the dataset selection. We first address the pivotal phase of data preprocessing. Prior to applying contrastive learning for the downstream task of gaze estimation, we also establish a baseline by training a supervised

gaze estimation model. Subsequently, we examine the integration of contrastive learning within the gaze estimation framework.

Chapter 5 – Eye contact detection: In this chapter, we explore the application of contrastive learning in the context of the downstream task of eye contact detection.To fine-tune the downstream model we not only used the ground truth of eye contact as a label, but also used gaze estimation to generate pseudo-labels

Chapter 6 – Discussion: In this chapter, we will discuss the results in chapters 4, 5, and 6. we aim to draw comprehensive insights and implications that contribute to the broader narrative of our study.

Chapter 7 – Conclusion

2 Background

This section provides an overview of the fundamental concepts and technologies that form the foundation of our research. We begin by discussing Artificial Neural Networks (ANN), which are the building blocks of modern deep learning and play a crucial role in various computational tasks. We then delve into Convolutional Neural Networks (CNNs), a specialized kind of ANN that has revolutionized the field of computer vision. A specific focus is given to ResNet, a CNN architecture known for its deep layers and efficiency in handling complex image recognition tasks.

Following this, we explore the concept of Contrastive Learning, a powerful technique in unsupervised machine learning. Contrastive Learning has gained prominence for its ability to learn useful representations by contrasting positive and negative examples, and it has become a crucial component in the development of robust and efficient models for tasks like gaze estimation and eye contact detection. In this section, we aim to shed light on how these technologies converge to create advanced models in the field of computer vision, particularly focusing on their application and significance in our research.

2.1 Artificial Neural Networks (ANN)

Artificial Neural Networks (ANNs)[LBH15], inspired by the neural structure of the human brain, replicate the function of biological neurons through a layered network of nodes. This network, illustrated in Figure 2.1, consists of an input layer, several hidden layers, and an output layer. The essence of an ANN is to learn the mathematical relationship between input and output, formally expressed as $y = f(x)$, where x represents the input and y the output.

In the training phase, ANNs are exposed to datasets comprising inputs and their corresponding expected outputs. Depending on the output type, the network either performs classification for discrete outcomes or regression for continuous ones. For instance, an ANN might classify a tree species from an image of a forest or predict a plant's height from its image.

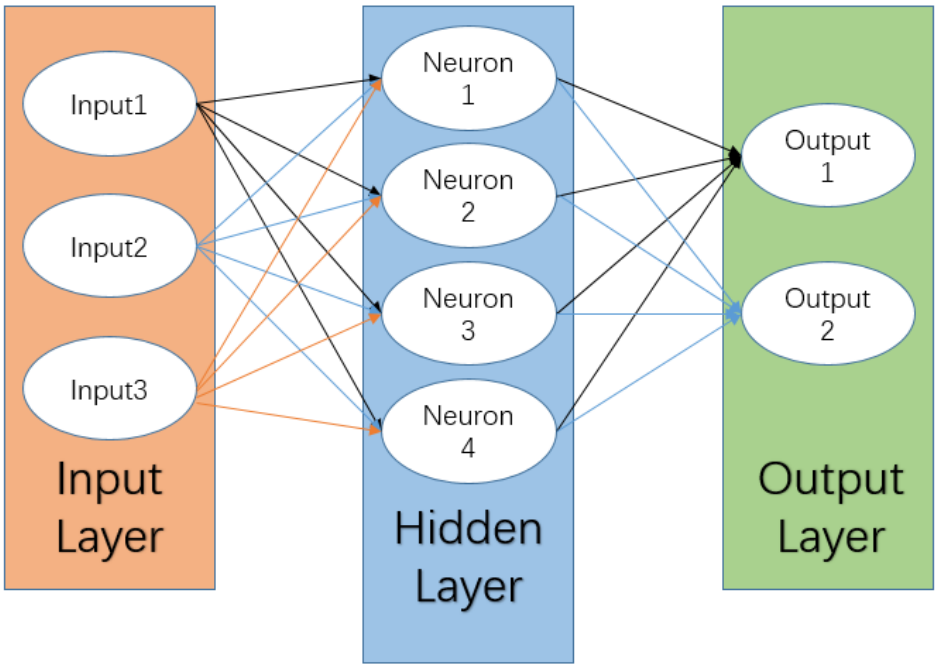

Figure 2.1: Artificial Neural Network

The training involves two primary steps. Initially, the network processes the input to produce a predicted output, which is compared against the expected output to calculate an error. This error serves as a feedback mechanism for learning.

Subsequently, in the backpropagation step, the error's gradient is utilized to adjust the network's internal parameters. This refinement enhances the accuracy of the input representation and improves the network's predictive capability. This iterative optimization process progressively improves the network's predictive accuracy.

ANNs offer several advantages over traditional machine learning methods. They autonomously extract relevant features from data, particularly useful for high-dimensional data analysis. They enable direct learning from raw inputs to outputs, eliminating the need for manual feature engineering or extensive domain knowledge. Moreover, ANNs can model complex non-linear data relationships, providing greater flexibility than conventional linear models.

2.1.1 Dense Layers

Dense layers, also referred to as fully connected layers, are integral components in neural networks. Picture each neuron in these layers as a basic processing unit. Every neuron

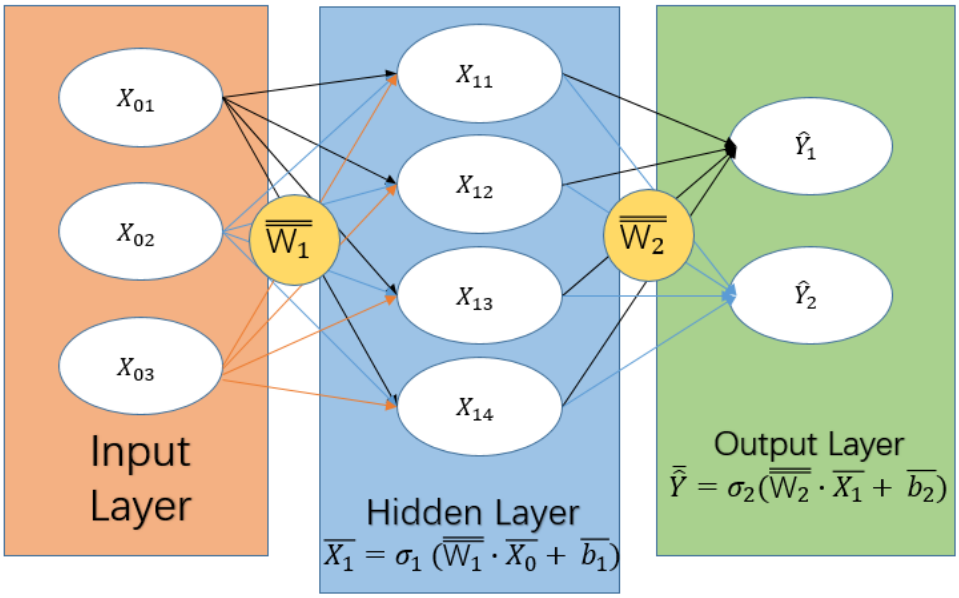

Figure 2.2: Fully Connected layer

connects to all the neurons in the layer before it. It takes incoming signals, multiplies them with specific values called weights, and then sums them up. This sum undergoes a transformation by an activation function, which determines what information should be passed forward. This mechanism enables the dense layer to filter and refine the input data, extracting meaningful patterns and information.

In practical applications, dense layers are highly adaptable. In tasks like image recognition, they typically follow initial processing layers. These first layers identify basic elements like edges or textures, and then the dense layers interpret these features to recognize more complex objects or shapes. Dense layers essentially combine simpler patterns from the earlier layers to understand and identify higher-level structures and concepts in the data. This process highlights their capability to integrate basic information into more complex and meaningful interpretations, crucial for advanced pattern recognition and data analysis tasks.

The formula for the output (y) of a neuron in a dense layer is given by:

$$y = \sigma \left(\sum_{i=1}^{n} w_i x_i + b \right) \tag{2.1}$$

Here: - y is the output of the neuron. - σ is the activation function. - w_i represents the weight associated with the input x_i. - b is the bias term. - n is the number of inputs.

This formula computes a weighted sum of the inputs, adds the bias, and then applies the activation function to produce the final output of the neuron.

2.2 Convolutional Neural Networks (CNN) and resnet

Convolutional Neural Networks, or CNNs[LBH15], are a type of neural network that's especially good at working with images, which sets them apart from traditional neural networks. Unlike regular Artificial Neural Networks (ANNs) that treat data as a flat line of information, CNNs are better at understanding images which are like grids of pixels. They do this by focusing on small parts of the image at a time, which helps them pick up on details like shapes and colors. This makes CNNs great for tasks that involve looking at and understanding pictures, such as identifying objects in photos or even creating new images.

CNNs use three main types of layers to do their work:

1. **Convolutional Layers:** These layers zoom in on small sections of the image to find specific features, like lines or corners.
2. **Pooling Layers:** These help to make the image data smaller and more manageable by summarizing the features found by the convolutional layers.
3. **Fully Connected Layers:** At the end, these layers take all the information the network has gathered and use it to make final decisions, like recognizing what's in the image.

Because of this specialized structure, CNNs are widely utilized in various image processing tasks, including identifying objects in photos, detecting different elements in an image, or even generating new images, demonstrating their effectiveness in understanding and interpreting complex visual information.

2.2.1 Convolutional Layers

Convolutional Neural Networks (CNNs)[LBH15] are a type of neural network optimized for detecting patterns in images. The convolutional layer is a key component of these networks, functioning differently from the dense layer that is commonly used in more basic neural network architectures.

Imagine the convolutional layer as equipped with a set of miniature flashlights, each spotlighting a small segment of the image. These flashlights seek out specific features like edges or textures within their illuminated patch. This contrasts with the dense layer, where every input is linked to every output, akin to a vast floodlight illuminating everything indiscriminately.

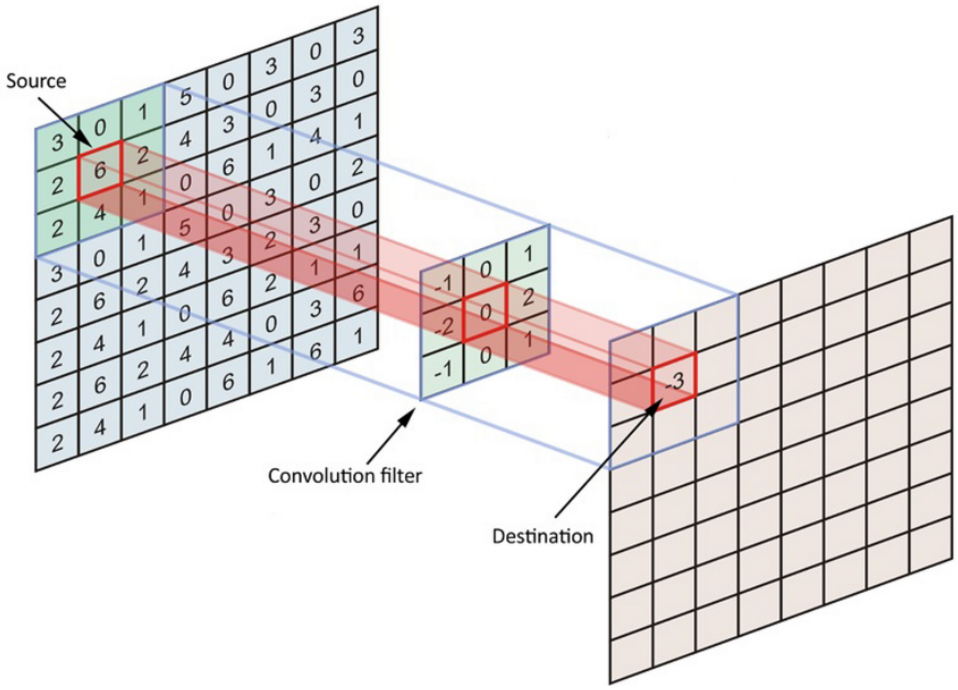

Figure 2.3: Convolution Operation[LIAP22]

The mathematics underpinning the movement of these flashlights, or kernels, across the image is articulated in the following equation:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{2.2}$$

In this equation, $S(i,j)$ represents the resulting value at a specific point in the image after analysis by the flashlight, I symbolizes the image under examination, and K denotes the pattern of the flashlight's filter.

To determine the size of the image post-analysis by all the flashlights, or the output dimension, the ensuing formula is employed:

$$O_{\text{dim}} = \frac{I_{\text{dim}} - K + (K-1)(D-1) + 2P}{S} + 1 \tag{2.3}$$

This formula contemplates the original image size (I_{dim}), the flashlight's beam size (K), the stride or step size when shifting the flashlight (S), the additional space around the image edges to ensure pattern detection there as well (P for padding), and the potential skipping of spaces by the flashlight (D for dilation).

The convolutional layer functions as a critical component in a neural network that examines small sections of an image to analyze and interpret its features. It uses defined mathematical operations to process the image data thoroughly and capture detailed

patterns within it. This systematic approach allows the network to effectively recognize various elements in the image and understand the overall content.

Introduction to Activation Functions:

Activation functions are vital in neural networks, enabling them to capture and represent nonlinear relationships within the data. By integrating these functions, neural networks can go beyond just understanding simple, linear connections to learning from the more complex, nonlinear interdependencies that are often present in real-world data. This nonlinear processing is key to the network's ability to discern intricate patterns and make predictions based on the rich and varied inputs it receives.

Here's a brief look at some common activation functions, along with their mathematical definitions:

- ReLU (Rectified Linear Unit): This function allows only positive values to pass through, effectively turning off the negative values:

$$f(x) = \max(0, x) \tag{2.4}$$

- Sigmoid: It compresses the input values within a range of 0 to 1, making it useful for models where we need to predict probabilities:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.5}$$

- Tanh (Hyperbolic Tangent): This one also squishes values but within a range of -1 to 1:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{2.6}$$

- Softmax: Often used in multi-class classification tasks, it calculates the probability for each class over all possible classes:

$$f_i(x) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{2.7}$$

- Leaky ReLU: A variant of ReLU that allows a small, nonzero gradient when the unit is not active:

$$f(x) = \max(\alpha x, x) \tag{2.8}$$

  The parameter α is a small constant that gives a slight slope to keep the updates alive even for negative input values.

Each function is tailored for specific scenarios: ReLU for general use due to its efficiency, sigmoid and tanh for predicting probabilities, Leaky ReLU to mitigate the "dying ReLU" problem, and Softmax for situations involving multiple categories.

2.2.2 Pooling Layers

Pooling layers in neural networks play a crucial role in downsampling and reducing the spatial dimensions of feature maps. This process is essential for enhancing the network's robustness to variations in input and, concurrently, reducing the computational complexity. The primary purpose of pooling is to retain vital information while decreasing the number of parameters, making the network more computationally efficient.

Typically performed after convolutional layers, pooling involves dividing the input into small regions and summarizing them. This can be achieved by taking either the maximum value (max pooling) or the average value (average pooling).

Max Pooling: Selects the maximum value from small input regions.Emphasizes the most prominent features, providing translation invariance and spatial dimension reduction.

Average Pooling: Computes the average value from small input regions.Provides a more generalized representation, reducing spatial dimensions while maintaining a smoother summary of features.

The downsampling achieved by pooling makes it an integral component in tasks such as image classification and object detection within the field of computer vision. Its ability to summarize information while reducing spatial dimensions contributes significantly to the network's overall efficiency.

2.2.3 Resnet

Before delving into the specifics of ResNet[HZRS16] architecture, it is important to highlight its significant role in our research. In this study, we extensively utilize ResNet18[HZRS16] (Figure 2.5) as the backbone architecture for our models. ResNet18, a variant of the ResNet family, is renowned for its efficiency and effectiveness in handling complex image processing tasks. As a common yet powerful CNN architecture, ResNet18 provides the necessary depth and complexity required for our applications, while maintaining a balance between computational efficiency and performance. The choice of ResNet18 is driven by its proven track record in achieving remarkable results in various computer vision tasks, making it an ideal choice for the foundation of our

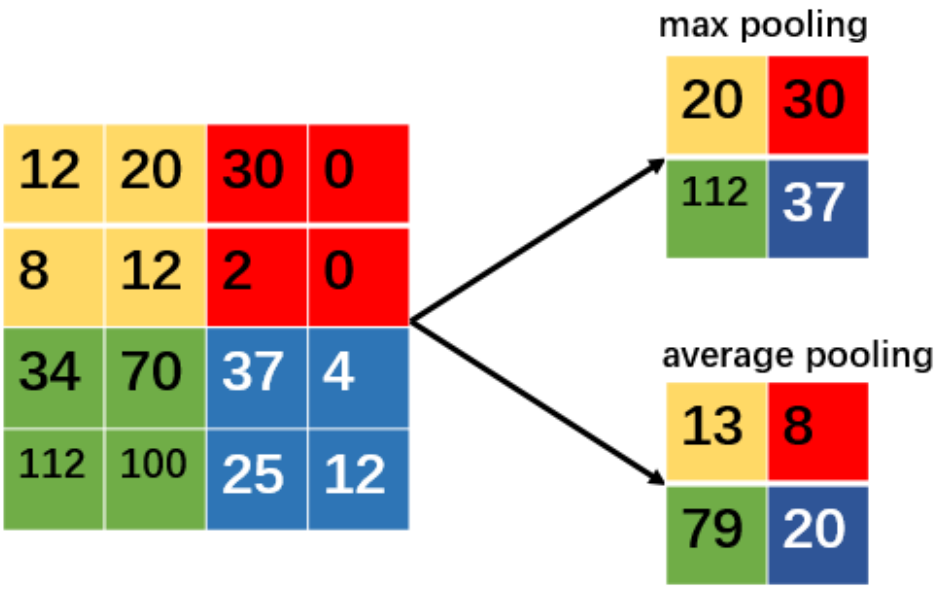

Figure 2.4: Pooling layers: the figure illustrates max pooling and average pooling. Max pooling selects the largest number from each colored grid section, while average pooling computes the average of the numbers in each section, simplifying the data representation.

advanced models. In the following sections, we will explore the ResNet architecture in more detail, underscoring its innovative features and why it stands out as a preferred choice in the field of deep learning and computer vision.

ResNet[HZRS16], short for Residual Networks, is a type of deep learning model that's been a game-changer for working with very deep neural networks. Its main innovation is the use of residual blocks that make it easier to train these deep networks without running into common problems like the vanishing gradient.

Here's a breakdown of what makes ResNet special:

- **Residual Blocks**: These are the core of ResNet and have something called shortcut connections that let the training process bypass some layers, which helps the gradient to flow through the network more effectively.
- **Skip Connections**: By jumping over some layers, these connections help the network to focus on learning the differences between the input and output of these blocks. This can make optimizing the network simpler and more effective.
- **Deep Architectures**: ResNets are known for their depth, with versions having hundreds or even thousands of layers, which has helped them perform exceptionally well in tasks like recognizing what's in an image.

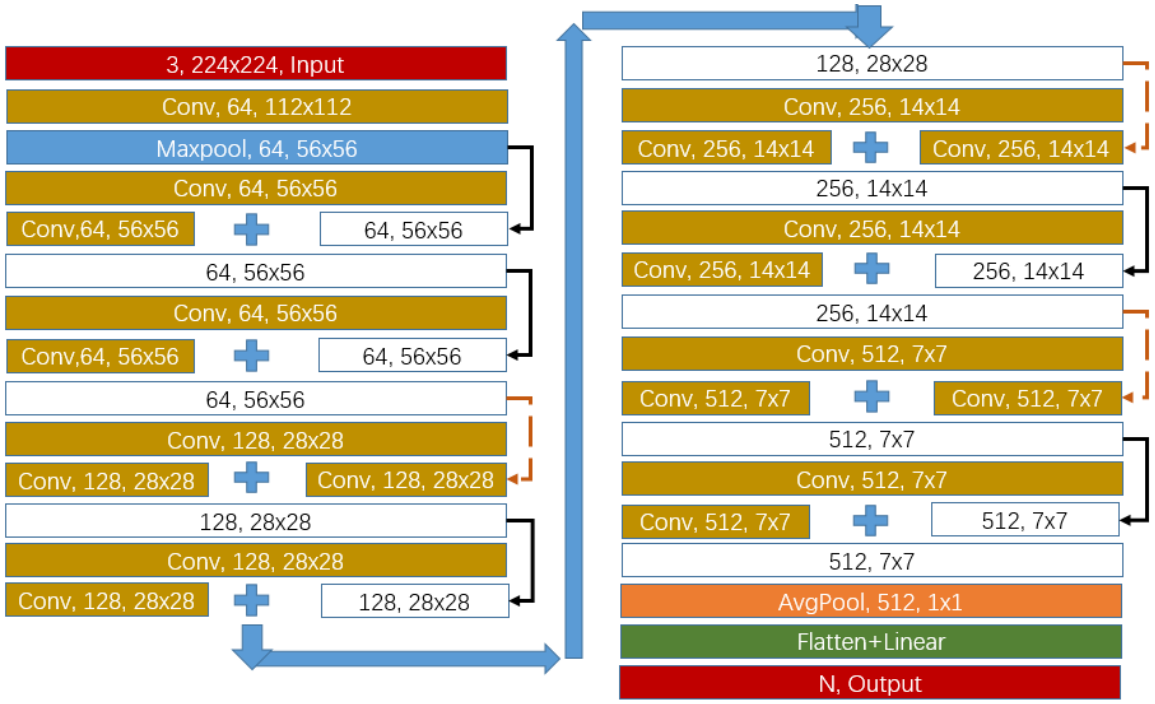

Figure 2.5: Resnet18

- **Global Average Pooling (GAP)**: At the end of the network, ResNet uses GAP instead of the usual fully connected layers. This reduces the dimensions to just one number per feature map, which cuts down on the number of parameters and helps avoid overfitting.

In terms of applications, ResNet has significantly advanced the field of computer vision. It's widely used for image classification, achieving remarkable accuracy improvements as documented by He et al. [HZRS16]. In object detection, ResNet has enhanced the precision of models like Faster R-CNN [RHGS15]. For image segmentation, which is crucial in areas like medical imaging, ResNet has improved the functionality of systems such as U-Net [RFB15]. These applications showcase ResNet's versatility and its substantial contribution to the progression of deep neural networks.

2.2.4 Optimization and Training

Training a neural network is all about Adjusting its parameters—weights and biases—to get the best possible predictions. The process revolves around an objective function, known as the loss function, which measures how far off the predictions are from the actual results. The training routine involves several steps using optimization algorithms that tweak these parameters to minimize the loss.

Here are the basic steps involved in neural network training:

1. **Initialization**: Begin with random values for the parameters or start from values learned previously by another model.
2. **Forward Propagation**: Feed the input data through the network to produce predictions.
3. **Loss Computation**: Evaluate how accurate the predictions are by comparing them to the true values using the loss function.
4. **Backward Propagation**: Calculate the gradient of the loss function with respect to each parameter. This is done by tracing the path back through the network (hence 'backpropagation').
5. **Parameter Update**: Adjust the parameters in a way that reduces the loss. This is usually done using optimization techniques like Stochastic Gradient Descent or Adam.
6. **Repeat**: Continue the process over multiple cycles, or epochs, to continuously improve the parameters.

For the optimization methods, here's a simplified explanation with the necessary mathematical details:

- **Stochastic Gradient Descent (SGD)**: It updates parameters using the loss gradient, guided by a learning rate. Mathematically, it's represented as:

$$\theta_{t+1} = \theta_t - \eta \nabla J(\theta_t) \tag{2.9}$$

  Here, θ is the parameter vector, η is the learning rate, and $J(\theta)$ is the loss function.

- **RMSprop**: This method adjusts the learning rate for each parameter based on the recent history of gradients, aiming to resolve the vanishing or exploding gradient problem. It's given by:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t + \epsilon}} \nabla J(\theta_t) \tag{2.10}$$

  with θ as the parameter, η the learning rate, v_t the moving average of the squared gradients, and ϵ a small number to ensure numerical stability.

- **Adam**: This optimizer combines momentum and RMSprop's approaches, automatically adjusting the learning rate for each parameter. The update rules are:

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) \nabla J(\theta_t) \tag{2.11}$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2)(\nabla J(\theta_t))^2 \tag{2.12}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_{t+1}} + \epsilon} m_{t+1} \tag{2.13}$$

In this, m_{t+1} and v_{t+1} are estimates of the first and second moments of the gradients, β_1 and β_2 are decay rates for these moments, and ϵ is a small constant for numerical stability.

By following these steps and using these optimization methods, neural networks can effectively learn from data, making them powerful tools for tasks like image and speech recognition, natural language processing, and many others.

2.3 Contrastive Learning

The primary goal of contrastive learning is to enhance model performance by capturing and understanding distinctions between samples. This method is widely applied in representation learning, particularly in unsupervised and self-supervised learning contexts.

Contrastive learning places a strong emphasis on obtaining concise representations of input data, aiming to ensure that similar samples are closely aligned in the feature space, while dissimilar ones are distinctly separated. In more concrete terms, this means that the model learns to map similar inputs to nearby points and dissimilar inputs to distant points in a high-dimensional space. This approach enables the model to understand and capture the essential patterns and relationships within the data, beyond the scope of a single specific task. As a result, it greatly enhances the model's ability to generalize, allowing it to perform effectively across a wide range of tasks and datasets by recognizing underlying data structures that transcend task-specific characteristics.

In practical terms, contrastive learning commonly involves creating pairs of positive and negative samples, where similar instances are labeled as positive and dissimilar ones as negative. During training, the model is optimized to minimize the distance between positive samples and simultaneously maximize the distance between negative samples.

Within the domain of contrastive learning, training often relies on various contrastive loss functions. Commonly used methods include Siamese networks [BGL+94], Triplet Loss [SKP15], and more advanced approaches like Contrastive Divergence Loss [Hin02]. These methods play a crucial role in guiding the learning process to ensure effective feature representation and discrimination between similar and dissimilar samples.

In summary, contrastive learning stands out as a powerful paradigm for training models with improved generalization capabilities, particularly in scenarios with limited labeled data.

2.3.1 SimCLR

Prior to delving into the specifics of SimCLR, it is essential to recognize its status as a concise yet effective framework within the realm of contrastive learning. In this paper, we place a significant emphasis on the SimCLR framework, particularly in its application to gaze estimation and eye contact detection tasks. Our approach involves adapting and optimizing SimCLR to better suit these specific tasks. One of the key strengths of SimCLR lies in its data augmentation component, which is versatile enough to incorporate various data augmentation methods. This flexibility, coupled with its straightforward architecture, provides considerable leeway for customization and optimization of the framework. Such adaptability makes SimCLR an ideal candidate for our research, allowing us to explore and enhance its capabilities in the context of advanced vision-based tasks.

SimCLR[CKNH20] is a framework designed for the unsupervised learning of visual representations, rooted in the concept of contrastive learning. The model aims to maximize the similarity between positive pairs (augmented views of the same image) while simultaneously minimizing the similarity between negative pairs (views from different images). Notably, SimCLR has demonstrated remarkable success in acquiring high-quality image representations, eliminating the dependence on labeled data.

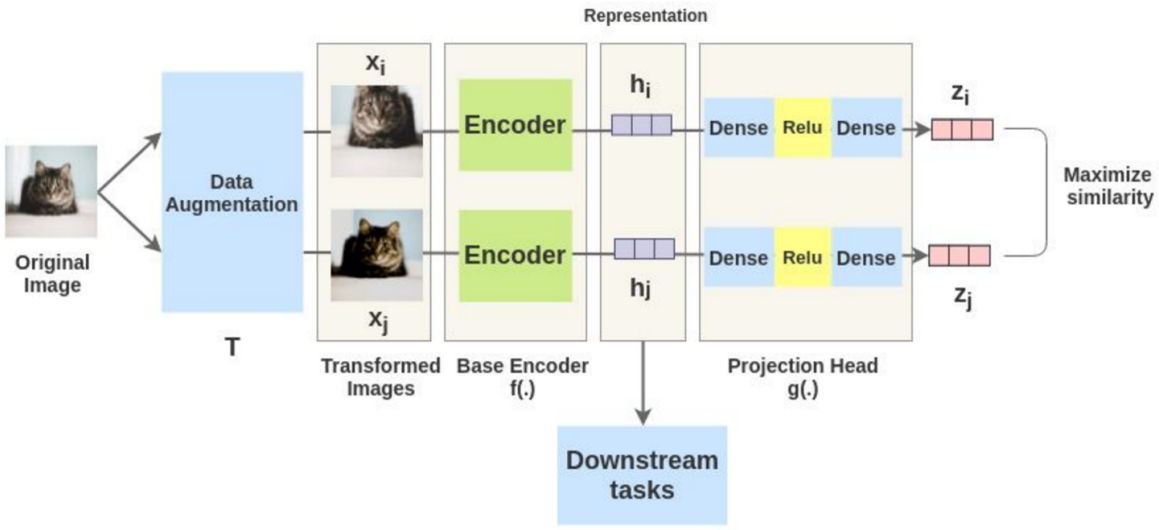

Figure 2.6: SimCLR[CKNH20]

SimCLR employs robust data augmentation techniques to generate diverse views of a single image. This strategy helps the model acquire invariant representations by promoting the recognition of consistent content across different transformations.

The training objective is characterized by a contrastive loss, specifically the Normalized Temperature-Scaled Cross-Entropy (NT-Xent) loss[CKNH20]. This loss is designed to narrow the distance between positive pairs in the feature space while simultaneously driving apart negative pairs.

For a given pair of samples x_i and x_j, where x_i is the anchor sample, x_j is the positive sample, and x_k is the negative sample, The Normalized Temperature-Scaled Cross-Entropy Loss (NT-Xent Loss)[CKNH20] is computed as follows:

$$\text{NT-Xent Loss}(x_i, x_j, x_k) = -\log\left(\frac{\exp\left(\frac{\text{sim}(f(x_i),f(x_j))}{\tau}\right)}{\sum_{n=1}^{N} \mathbb{1}_{[n\neq i]} \exp\left(\frac{\text{sim}(f(x_i),f(x_n))}{\tau}\right)}\right) \tag{2.14}$$

where:

- $f(\cdot)$ represents the embedding function of the model, mapping inputs to the representation space.
- $\text{sim}(a, b) = \frac{a \cdot b}{\|a\|\|b\|}$ is the cosine similarity function, measuring the similarity between two samples.
- τ is the temperature parameter, adjusting the scale of the similarity distribution between samples.
- $\mathbb{1}_{[n\neq i]}$ is the indicator function, equal to 1 when $n \neq i$ and 0 otherwise.
- N is the size of the sample set.

The objective of this loss function is to maximize the similarity between positive samples while minimizing the similarity between negative samples.

SimCLR utilizes a neural network architecture, commonly a deep convolutional neural network (CNN), acting as an encoder to transform input images into high-dimensional feature vectors. The encoder is trained to extract representations that capture meaningful and semantically rich information.

SimCLR incorporates a projection head designed to map high-dimensional representations to a lower-dimensional space. This addition significantly contributes to enhancing the quality of the learned representations.

During the training process, positive and negative pairs are sampled from augmented views. The model is subsequently optimized to maximize agreement between positive pairs while minimizing agreement between negative pairs. This iterative process fosters the learning of a representation space where similar samples are proximate, and dissimilar samples are distanced.

Training Procedure:

1. **Positive Pair Generation:** For each image, multiple augmented views are created through data augmentation. These views form positive pairs.

2. **Negative Pair Sampling:** Negative pairs are sampled by selecting views from different images in the batch.

3. **Contrastive Loss Computation:** The NT-Xent loss is computed based on the cosine similarity between positive pairs and the negative pairs.

4. **Optimization:** The model parameters are updated using gradient descent to minimize the contrastive loss.

Benefits:
- SimCLR has demonstrated remarkable success in self-supervised learning, achieving state-of-the-art results in various computer vision tasks.
- The learned representations generalize well to downstream tasks with limited labeled data.

In our discourse on SimCLR, a prominent model in contrastive learning, we now turn our attention to the computation of Top-1 Accuracy. This metric is pivotal in assessing the model's ability to correctly discern positive pairs from a set of negatives.

SimCLR operates on the principle of maximizing agreement between different augmentations of the same data instance in a latent space, leveraging a contrastive loss. Top-1 Accuracy, in this context, quantifies the model's efficiency in this representational learning task.

Computation of Top-1 Accuracy:

1. **Embedding Generation**: For each data instance in a batch, SimCLR generates embeddings, including two distinct augmentations of the same image (positive pairs) and other disparate images (negatives).

2. **Pairwise Distance Calculation**: The model computes the distance between the embedding of each anchor and every other instance in the batch, typically using cosine similarity.

3. **Identification of Nearest Embedding**: For each anchor, the nearest embedding in the latent space is identified.

4. **Calculation of Top-1 Accuracy**: The Top-1 Accuracy is determined by the proportion of instances where the nearest embedding to an anchor is its associated positive pair. Formally, it is defined as:

$$\text{Top-1 Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left\{\text{argmin}_{j\neq i}\, d(\mathbf{z}_i, \mathbf{z}_j) = \text{Positive Pair of } \mathbf{z}_i\right\}$$

   where N represents the number of instances in the batch, $\mathbf{z}_i$ and $\mathbf{z}_j$ are the embeddings of the anchor and other instances, $d(\cdot)$ denotes the distance metric, and $\mathbb{1}\{\cdot\}$ is the indicator function, equating to 1 when the argument is true, and 0 otherwise.

In essence, Top-1 Accuracy in SimCLR is a critical measure reflecting the model's prowess in distinguishing similar from dissimilar instances through learned representations.

3 Related Work

As the focus of this thesis is on the application of contrastive learning to the task of eye contact detection, this section will provide an overview of the traditional methods employed to address eye contact detection. Additionally, we will explore gaze estimation, a task closely related to eye contact detection. There has been noteworthy work in gaze estimation that employs both traditional approaches and contrastive learning methods. Insights from these studies are particularly relevant and have informed the methodologies adopted in this thesis. By examining the existing literature, we aim to contextualize our research within the broader landscape of eye contact and gaze-related tasks and highlight the contributions of contrastive learning in this domain.

3.1 Appearance-based Gaze Estimation

The realm of gaze estimation has witnessed remarkable growth, with recent innovations emphasizing the significance of harnessing comprehensive facial cues to elevate model accuracy. Pioneering this evolution, Zhang et al. introduced a groundbreaking full-face approach [ZSFB17a], a departure from the eye-centric methods of yesteryears, by utilizing the entire visage as the input to convolutional neural networks (CNNs). Their technique strategically employs spatial weights to enhance or diminish information across the facial feature maps, fine-tuning the model's focus.

This method laid the groundwork for subsequent research by Smith et al. [SXYZ17b] and Lee et al. [LYZX18c], who incorporated head pose and eye appearance as auxiliary indicators, further refining gaze direction predictions. These contributions signal a paradigm shift in gaze estimation, moving towards a more integrative approach that considers the face not just as a collection of features, but as a dynamic map where context and subtle cues converge.

In a similar vein, the works of Patel and Smith [PS19d] have leveraged the prowess of deep learning to automate the extraction of pertinent features from the facial data, bypassing the laborious process of manual feature selection. It's these advancements that epitomize the trend towards comprehensive, context-sensitive models in gaze

estimation—a trend that now beckons the adoption of contrastive learning techniques to further distill and enhance the feature extraction process.

The trajectory of gaze estimation research, from its initial eye-focused methods to the incorporation of full-face and head pose data, charts a course of discovery and innovation.

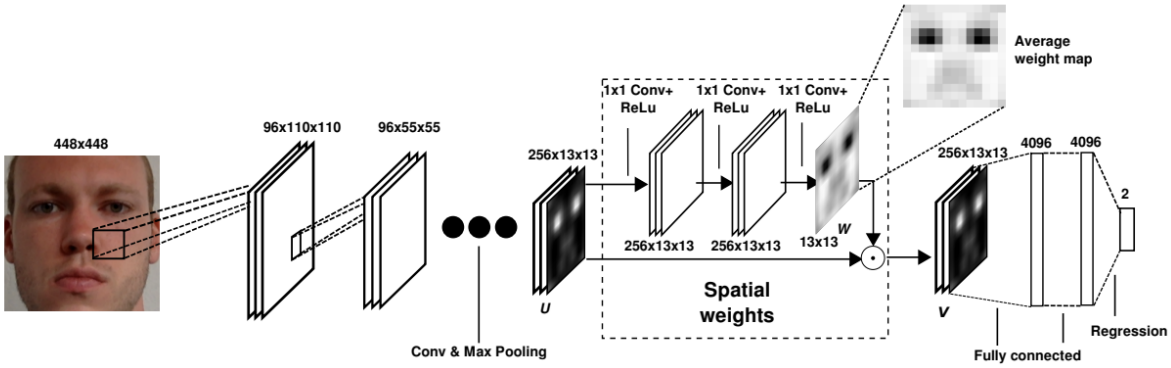

Figure 3.1: Full-face appearance-based gaze estimation[ZSFB17a]

3.1.1 Contrastive Learning for Appearance-based Gaze Estimation

While original appearance-based gaze estimation methods operate on supervised learning principles and depend on extensive datasets, there is a growing body of research exploring the application of contrastive learning methods to the appearance-based gaze estimation task. These methods demonstrate the ability to achieve comparable performance using a significantly smaller number of datasets compared to original supervised learning models.

GazeCLR[JM22]

In the first training phase of GazeCLR, the framework employs a contrastive learning approach to train a shared encoder. Distinct from SimCLR, GazeCLR incorporates both single-view and multi-view comparison methods. To facilitate these different learning objectives, GazeCLR integrates two separate MLP-based projection heads post-encoder: one dedicated to learning invariance through single-view comparison and the other for learning equivariance via multi-view comparison.

The construction of positive and negative pairs in GazeCLR is meticulously orchestrated as follows:

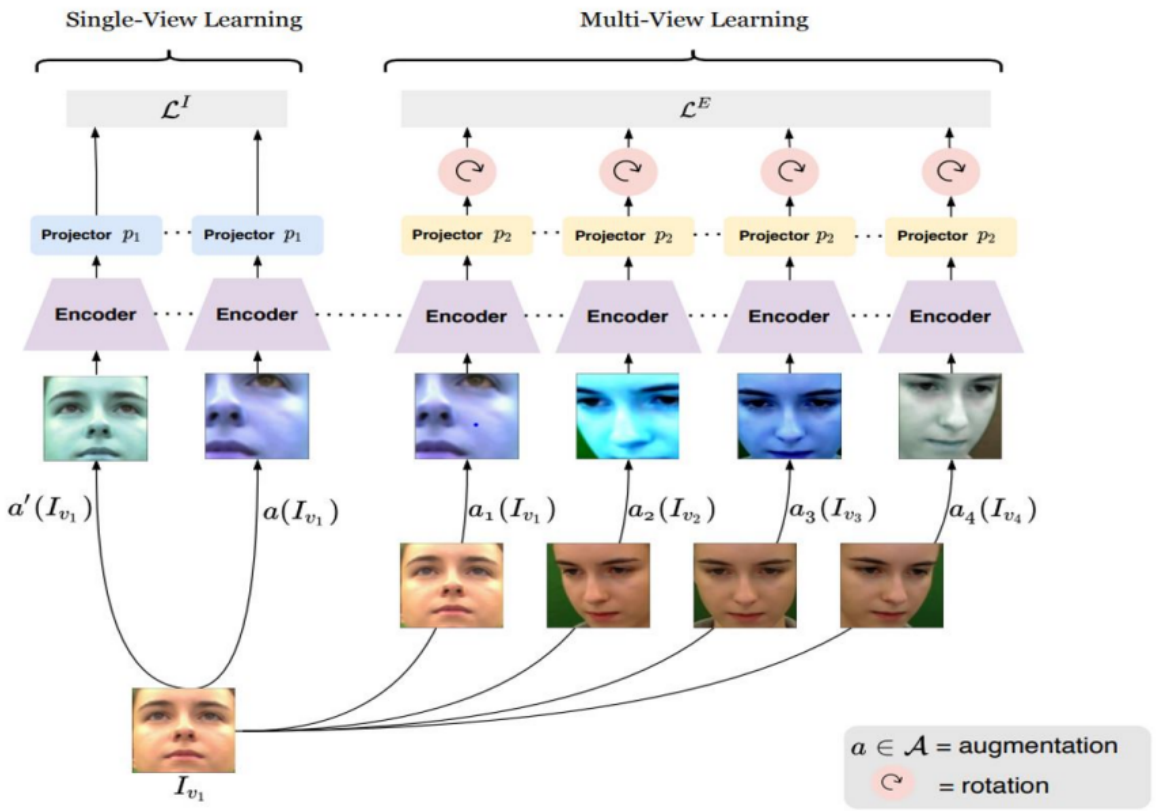

Figure 3.2: Overview of the GazeCLR architecture. The input image is processed through data augmentations to generate multiple views, which are then fed into a shared encoder in both single-view and multi-view learning settings. In single-view learning, variants of the same image are treated as positive pairs, while in multi-view learning, augmented images from different perspectives of the same subject are paired as positive pairs. Fgiure taken from:[JM22]

- **Single-View Positive Pairs:** The framework selects an image $I_{vi,t}$ from a specific camera view vi at a given timestamp t. It then applies two distinct augmentations from a predefined set of appearance transformations A, creating positive pairs to facilitate invariance learning.
- **Multi-View Positive Pairs:** In this approach, GazeCLR considers all unique pairs of camera viewpoints (vi, vj) at the same timestamp t. Images from these different camera views are selected and augmented with transformations from A to induce equivariance learning.

For negative pairs, GazeCLR does not require explicit sampling; instead, it treats all other images within the mini-batch as negative examples.

The first phase of training within the GazeCLR framework, focusing on the shared encoder, does not necessitate labeled datasets. The encoder, trained to extract features pivotal for gaze estimation, is later fine-tuned in a subsequent phase. This second phase involves appending an MLP-based regressor head to the pre-trained encoder, allowing for efficient fine-tuning using a smaller annotated dataset for precise gaze estimation, ultimately leading to the accurate prediction of 3D gaze directions.

In summary, GazeCLR represents a significant adaptation of the SimCLR contrastive learning framework, specifically tailored for the gaze estimation task. This adaptation underlines the feasibility of modifying SimCLR to better suit specific tasks, inspiring the exploration undertaken in this thesis. While GazeCLR demonstrates the potential of such optimizations for gaze estimation, its application and performance in the realm of eye contact detection remain unexplored. This thesis seeks to bridge this gap, investigating how similar adaptations to SimCLR, particularly in its data augmentation and pair construction methodologies, could enhance its applicability and effectiveness in eye contact detection tasks.

3.2 Eye Contact Detection

In light of our focus on contrastive learning for eye contact detection tasks, we particularly review research that utilizes deep learning techniques to tackle the problem of eye contact detection. The GazeLocking method by Smith et al. [SYFN13] is a noteworthy example, employing classification techniques to detect eye contact between a person and a camera. Ye et al. extend this line of inquiry with a wearable camera system that detects eye contact from the wearer's perspective, leveraging a supervised learning framework [YLL+15]. Differing from these perspectives, Recasens et al. [RKVT15][RVKT16] approach the problem from a third-person viewpoint, where the camera captures both the subject and the gaze target within the scene, utilizing a CNN-based model to predict the focus of gaze.

However, these methods share limitations with conventional Appearance-based gaze estimation approaches, such as the need for user-specific or environment-specific training. They also often rely on prior knowledge about the target object's size and location. Addressing these limitations, Zhang et al. [ZSB17] introduced an unsupervised method, unique for its ability to adapt to specific camera-target settings by collecting relevant training data on-site, thus enhancing the flexibility and applicability of eye contact detection systems.

Unsupervised eye contact detection

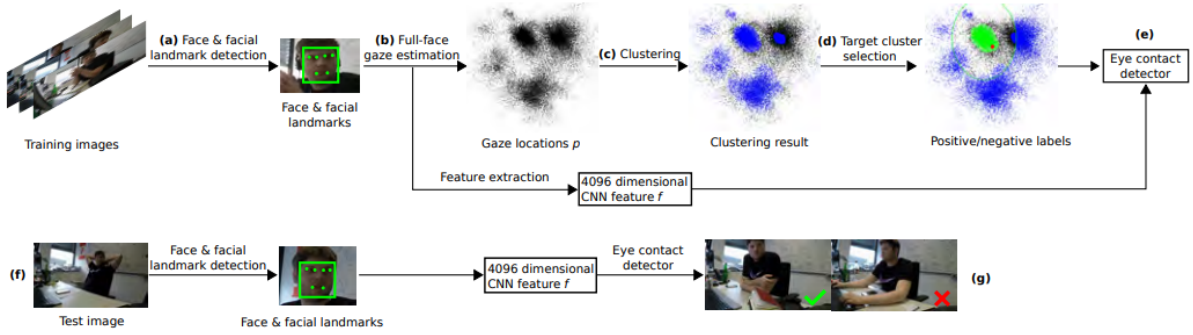

Figure 3.3: Unsupervised eye contact detection[ZSB17]

The proposed method, introduced by Zhang et al.[ZSB17], is centered on unsupervised eye contact detection utilizing a single off-the-shelf RGB camera positioned near the target object. In the training phase, the approach involves face and facial landmark detection, followed by the application of a state-of-the-art full-face appearance-based gaze estimation method. Subsequently, the estimated gaze directions undergo clustering, and the cluster corresponding to the target object is identified. This clustering result is then used to label samples for positive and negative eye contact. A two-class SVM is trained on high-dimensional features extracted from the gaze estimation CNN.

During the gaze estimation phase, a comprehensive full-face method is employed, trained on diverse datasets to handle variations in illumination, head pose, and gaze direction. This method leverages face detection, facial landmark detection, and data normalization techniques. The gaze estimation results are utilized for sample clustering, and a 4096-dimensional face feature vector is extracted for further analysis.

The sample clustering process involves rejecting unreliable samples based on facial landmark alignment scores and using them as negative samples during training. The OPTICS algorithm[ABKS99] is employed for clustering, and the cluster closest to the camera position is selected as the positive cluster corresponding to the target object. A safe margin around the positive cluster helps filter out noise samples.

Eye contact detection is performed using a two-class SVM classifier trained on labeled samples. To address potential class imbalance, a weighted SVM classifier is utilized. High-dimensional features extracted from the gaze estimation CNN are processed through PCA for dimensionality reduction during training. During testing, the SVM classifier predicts eye contact labels based on input features.

In conclusion, while Zhang et al.'s method [ZSB17] provides a foundational framework for unsupervised eye contact detection using a simple RGB camera, it has limitations.

The main constraint lies in the method's reliance on a supervised gaze estimation model to extract features for eye contact detection, resulting in suboptimal generalization for this specific task. Addressing this limitation, the current thesis proposes enhancements using contrastive learning models. Nevertheless, Zhang et al.'s work forms the basis of this thesis, with their experimental setup serving as the baseline for all eye contact detection experiments conducted herein.

4 Appearance-based Gaze Estimation

4.1 Introduction

Gaze estimation is a computer vision task that involves determining the direction in which a person is looking. The goal is to infer the point on a screen, scene, or image that corresponds to the person's gaze. This task has applications in various fields, including human-computer interaction, virtual reality, and driver monitoring systems.[Soc17]

Gaze estimation can be formally described as follows:

Let P represent the set of all possible gaze points in a given space, such as the pixels on a screen or the coordinates in a scene. For a specific individual, their gaze can be denoted as a vector $G = (x, y, z)$ in a three-dimensional space, where x, y, and z represent the horizontal, vertical, and depth components of the gaze, respectively.

Given an input image I containing the person's eyes or face, the task of gaze estimation is to predict the gaze vector G based on the visual information extracted from the image. This is typically formulated as a regression problem, where a model M is trained to map input images to corresponding gaze vectors:

$$G' = M(I) \tag{4.1}$$

where G' is the predicted gaze vector.

The accuracy of gaze estimation is often evaluated by measuring the angular error or Euclidean distance between the predicted gaze vector G' and the ground truth gaze vector G for a given set of test images.

In summary, gaze estimation is the process of predicting the direction of a person's gaze in a specified space based on visual input, and it is commonly formulated as a regression problem in computer vision.

4.2 Datasets

4.2.1 MPIIFaceGaze Dataset:

The MPIIFaceGaze dataset[ZSFB17a], initially developed for supervised full-face gaze estimation, includes detailed annotations for 37,667 images from 15 participants. It encompasses a wide range of gaze-related data such as facial landmarks, head pose, and 3D gaze direction. While this dataset was initially explored in our research for its suitability in gaze estimation, challenges related to image quality and gaze angle distribution led us to switch to the GazeCapture dataset.[KKK+16] GazeCapture, with its higher image quality, broader gaze angle distribution, and mobile device-based image capture, aligns more closely with the requirements and context of our study.

4.2.2 GazeCapture Dataset:

The GazeCapture dataset[KKK+16] is a large-scale dataset used in gaze estimation research, featuring over 2.4 million images from more than 2,000 participants. This dataset is notable for its diversity in participant demographics and environmental conditions. Data was gathered using a mobile app, with selfies captured by the front-facing camera, making it well-suited for real-world applications. Each image includes precise 3D gaze direction annotations. Its wide range of gaze angles and rich participant diversity make GazeCapture particularly relevant for our study's mobile device context.

However, the MPIIFaceGaze and GazeCapture datasets lack eye contact detection labels. In response to this limitation, our project proposes an unsupervised contrastive learning framework. This framework aims to extract more generalized features from the unlabeled GazeCapture dataset, applying these features to the task of eye contact detection.

4.3 Data Pre-processing

4.3.1 Data Normalization

In the realm of appearance-based gaze estimation, some methods assume a frontal head pose. However, real-world settings frequently involve head rotations. Appearance-based gaze estimators aim to address the challenge of accurately inferring 3D gaze directions, regardless of the initial appearance of the faces in input images.

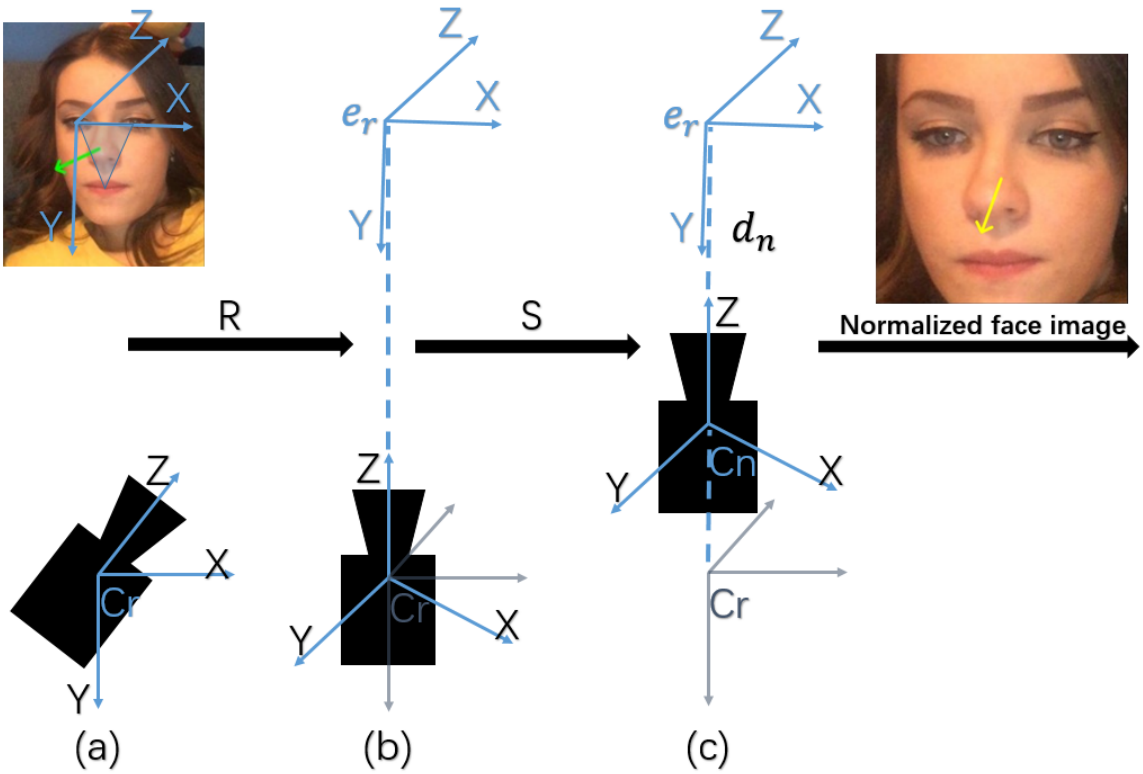

Figure 4.1: The figure illustrates the stages of eye image normalization. Initially, the head pose coordinate system, centered at the eye's center (er), has an arbitrary orientation with respect to the camera's coordinate system. In step (b), the camera coordinate system is rotated by applying a rotation matrix (R) to align with the eye center. Finally, in step (c), the world coordinate system is scaled using a scaling matrix (S), leading to a normalized face image where the eye's position is uniformly represented, facilitating accurate gaze estimation.

Additionally, it is noteworthy that variations in the scale or distance of the face impact eye appearance. Different distances between the camera and the eye result in varying sizes of the eye in captured images. Furthermore, the non-planar nature of the eye introduces alterations in its appearance under different viewing conditions.

Data normalization is crucial in learning-based gaze estimation, serving to mitigate variations stemming from head rotation and translation. The section outlines the data normalization process, initially introduced by Sugano et al. (2014)[SMS14] and Zhang et al. (2018)[ZSB18]. The narrative then shifts to address a specific issue encountered in handling 2D images, leading to the introduction of a modified normalization method.

This method incorporates a more robust planarity assumption, aimed at resolving the identified issue.

The data normalisation process introduced by Sugano et al. (2014)[SMS14] has the following three steps:

1. Leveraging face detection and facial landmark detection methods to identify landmarks in images captured by a calibrated monocular RGB camera, we improve upon the foundational work of Sugano et al. [SMS14]. Utilizing the latest in technology advancements, the insightface model [Dee23] is employed for its superior accuracy in both detecting faces and pinpointing facial features. Should multiple detections occur, the largest bounding box is selected. Any images without detectable faces are excluded. Subsequent facial landmark detection through the insightface model [Dee23] further refines the process, providing a robust approach for our research.

2. The approach utilizes a generic 3D facial shape model F for estimating the 3D pose of detected faces from 2D facial landmarks for gaze estimation [ZSFB17b]. This model, constructed by averaging across all participants, includes the 3D positions of significant facial landmarks. A right-handed coordinate system is established with the x-axis spanning the eye midpoints, the y-axis perpendicular within the eye-mouth triangle, and the z-axis orthogonal to the face plane. The Perspective-n-Point problem, solved via the EPnP algorithm and refined by the Levenberg-Marquardt method, determines the 3D rotation matrix R_r and translation vector t_r, providing a robust solution for 3D pose estimation in practical gaze estimation scenarios.

3. After initial face detection, landmark localization and 3D pose estimation, the face image normalization process [SMS14] commences. This involves aligning the head coordinate system (HCS) to the camera coordinate system (CCS), where HCS is defined by facial landmarks. The normalization ensures the camera's alignment with the HCS origin, co-planarity of x-axes, and consistent eye size at a fixed distance.

   The rotation matrix R aligns the CCS's z-axis with the HCS's origin and ensures the x-axes are co-planar. The scaling matrix S is defined to maintain the eye size across all normalized images:

$$R = \begin{bmatrix} \frac{\mathbf{x}_c}{\|\mathbf{x}_c\|} \\ \frac{\mathbf{y}_c}{\|\mathbf{y}_c\|} \\ \frac{\mathbf{z}_c}{\|\mathbf{z}_c\|} \end{bmatrix}, \quad S = \text{diag}(1, 1, \frac{dn}{\|er\|}) \tag{4.2}$$

   Here, $\mathbf{x}_c$, $\mathbf{y}_c$, and $\mathbf{z}_c$ are the unit vectors of the camera's coordinate system in step(c) Figure 4.1 along the x, y, and z directions, respectively. dn represents the

fixed distance from the camera to the eye center, and $\|er\|$ is the magnitude of the translation vector from the camera to the eye center.

The total transformation matrix $M = SR$ applies to 3D face meshes directly. For 2D images, perspective warping uses $W = CnMC_r^{-1}$, with Cr and Cn being the original and normalized camera projection matrices, respectively.

Modified Data Normalization[ZSB18]:

In addressing the geometric transformations associated with eye image normalization, two distinct approaches are employed for 3D and 2D data.

For 3D data, the transformation of the gaze vector utilizes the transformation matrix M, defined as $M = SR$ where $S = \text{diag}(1, 1, \frac{d_n}{\|\mathbf{e}_r\|})$. The normalized gaze vector $\mathbf{d}_n$ is calculated as follows:

$$\mathbf{d}_n = M\mathbf{d}_r \tag{4.3}$$

This equation represents the transformation of the gaze direction in a 3D coordinate system through both rotation and scaling.

On the other hand, for 2D images, a modified approach is proposed[ZSB18]. While the image normalization process still employs the transformation matrix W defined as $W = C_nMC_r^{-1}$, the treatment of the gaze vector differs. In this context, only the rotation matrix R is applied to the gaze vector, omitting the scaling effect on the 3D coordinate system. This results in a different formulation for the normalized gaze vector:

$$\mathbf{d}_n = R\mathbf{d}_r \tag{4.4}$$

In this scenario, R is derived as RR_r, where RR_r represents the composition of the rotation matrices from both the 3D facial pose estimation and the normalization process. This adjustment ensures that the scaling is applied specifically to the camera projection matrix C_r in the context of 2D images. Additionally, the transformation is also utilized to reproject the estimated gaze vector back to the original camera coordinate system, expressed mathematically as $\mathbf{d}_r = R^{-1}\mathbf{d}_n$.

These mathematical formulations highlight the nuanced approach required for gaze vector normalization in both 3D and 2D data, ensuring accurate gaze estimation under varying conditions.

4.4 Supervised Appearance-based Gaze Estimation

Supervised appearance-based gaze estimation relies on supervised learning, utilizing labeled datasets that include both facial images and corresponding gaze directions for training. The method emphasizes estimating gaze from facial appearance features like eye regions, head pose, and facial expressions.

In a supervised context, the model minimizes the difference between predicted and ground truth gaze directions during training. The efficacy of this approach heavily relies on the volume and quality of the labeled dataset, underscoring the crucial dependence on substantial and high-quality data with accurate annotations for effective model performance.

4.4.1 Full-Face Gaze Estimation with a Spatial Weights CNN

The Full-Face Gaze Estimation method, introduced by Zhang et al.[ZSFB17a], addresses the core challenges of 2D and 3D gaze estimation, focusing on learning the regression function f.

The approach is rooted in the idea that beyond the eyes, other facial regions may contain valuable information for gaze estimation. To leverage information from the entire face, a Convolutional Neural Network (Spatial Weights CNN) is proposed for 2D and 3D gaze estimation. Additional layers are introduced in the final convolutional layer to learn spatial weights. The motivation behind spatial weighting lies in suppressing activations from non-contributing image regions, such as the background, to enhance performance. Moreover, activations from facial regions other than the eyes are expected to be subtle, varying based on input conditions like head pose, gaze direction, and illumination. To explicitly guide the network in learning the varying importance of different facial regions for gaze estimation, concepts from [ZSFB17a] involving three 1×1 convolutional layers and rectified linear unit layers are adapted for full-face gaze estimation.

Specifically, a single heatmap is generated to encode the overall importance across the entire face image, followed by element-wise multiplication with the feature map of the preceding convolutional layer.

It is essential to note that this method is supervised, and in this paper, we refer to this model as the Original Gaze Estimation model. The model is trained on the GazeCapture and MPIIFaceGaze datasets, serving as a baseline in our study.

4.5 Contrastive Learning for Gaze Estimation

Contrastive Learning for Gaze Estimation deviates from Supervised Appearance-based Gaze Estimation in its learning paradigm. Unlike the reliance on labeled datasets with explicit gaze directions, contrastive learning aims to learn by contrasting positive and negative samples within the dataset.

Following this, I will provide a detailed overview of how the contrastive learning model is specifically applied to gaze estimation. *Backbone* can be succinctly described as the part of the model that functions as a feature extractor, learning to distill the essential characteristics from the input data.

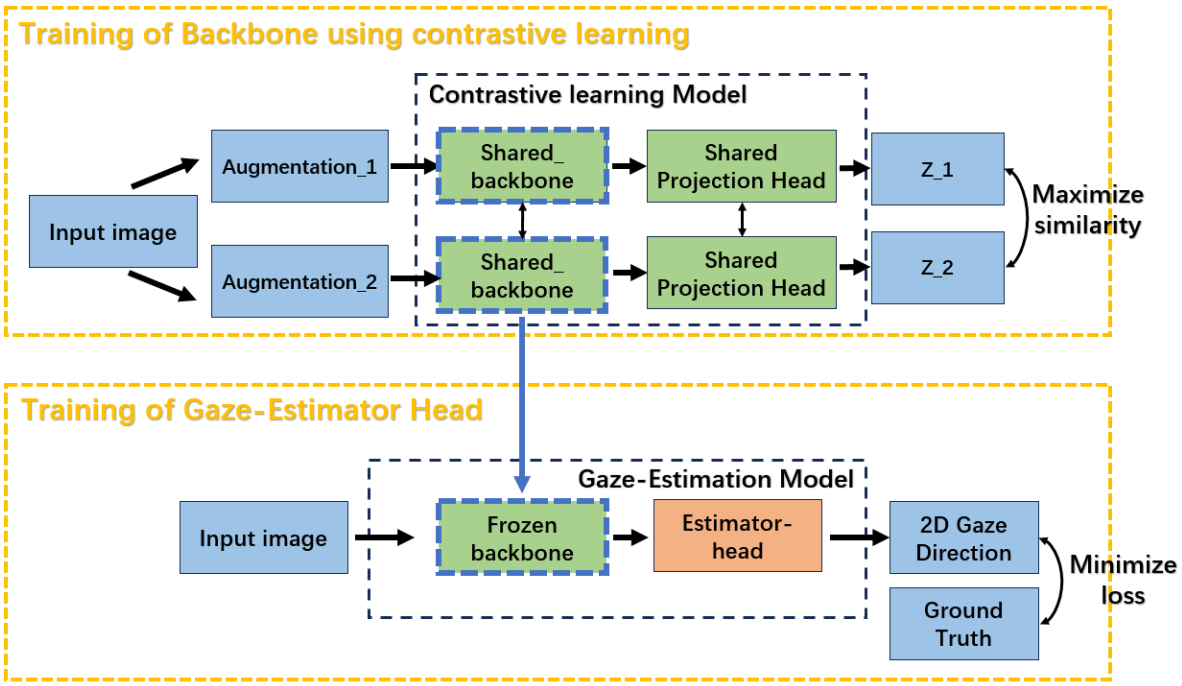

Figure 4.2: The Figure presents a Pipeline for training a gaze estimation model: the top section for training feature extraction through contrastive learning, and the bottom for refining gaze estimator head.

The diagram in Figure 4.2 illustrates the application of contrastive learning to gaze estimation in a two-stage process:

1. **Training of Backbone using contrastive learning:**
    - An input image is processed with two different augmentations to create two distinct versions of the same image.

- Both augmented images are fed through the same neural network, termed the "Shared backbone", which is responsible for feature extraction.
- After the shared backbone, each set of features is transformed by a "Shared Projection Head" into vectors z_1 and z_2.
- The objective in this stage is to maximize the similarity between z_1 and z_2, notwithstanding the initial augmentations. This trains the model to recognize the same object or scene irrespective of variations such as perspective or lighting.

2. **Training of Gaze-Estimator Head:**
   - The same input image from the first stage is passed through the shared backbone once more, which is now "frozen", indicating its weights do not update during this stage.
   - The extracted features by the frozen backbone are directed to a "Gaze-Estimator Model" with a specific "Estimator-head" designed for gaze direction estimation.
   - The estimator-head is tasked with predicting the 2D gaze direction, which is then evaluated against the ground truth to minimize the loss, quantifying the discrepancy between the predicted and true gaze directions.

By employing contrastive learning for the shared backbone training, the model acquires robust feature representations, which are advantageous for the gaze estimation task. The gaze estimator head utilizes these features to accurately predict the gaze direction in the input image. Contrastive learning enhances the model's generalization by learning to focus on consistent features across different augmentations.

For Gaze Estimation tasks, contrastive learning employs distinct methods when constructing positive and negative sample pairs, deviating from the conventional approaches used in general contrastive learning models. This specialized approach aims to focus the model specifically on features relevant to gaze direction. Two noteworthy strategies in this context include Gaze-specific data augmentation and the subject-conditional projection module[DZL23].

By incorporating these strategies, Contrastive Learning for Gaze Estimation offers a self-supervised learning framework that effectively captures gaze-related features without relying on explicit gaze annotations. This proves advantageous in scenarios where acquiring labeled gaze data is challenging or expensive, providing an alternative learning approach based on the intrinsic structure of the data.

4.5.1 SimCLR model with the gaze-specific data augmentation

To effectively train contrastive models for gaze estimation, it is imperative to recognize the limitations inherent in conventional data augmentation methods and to adapt these techniques to better suit the unique requirements of this task. Standard augmentations such as cropping, cutout, rotation, and flipping often fail to maintain essential features necessary for gaze analysis, specifically the eyes and the direction in which they are looking.

These typical approaches might remove or alter critical visual cues that are necessary for identifying the gaze direction, leading to the introduction of noise in the dataset. For instance, cropping and cutout can inadvertently eliminate the eyes from an image, which are fundamental for gaze estimation. Similarly, operations like rotation and flipping could modify the gaze direction itself, thereby corrupting the consistency required across augmented images (positive pairs) for effective learning.

This understanding underlines the necessity for gaze-specific data augmentation strategies. Such strategies would ensure the preservation of important gaze-related features and maintain gaze-semantic consistency across all transformed views of the same image. By refining data augmentation to protect against the loss of gaze information, we can train models that are not only robust to variations in image presentation but also attuned to the subtleties of gaze orientation. This optimization is key to advancing the performance of gaze estimation systems.

To achieve the aforementioned objectives, we have implemented a Gaze-specific Data Augmentation (GDA) technique[DZL23], meticulously tailored for gaze estimation tasks. This approach is designed to address and overcome the limitations of traditional augmentation methods specifically in the context of gaze estimation.

Building on the advantages of the GDA technique, the following section details the creation of a dedicated Gaze-Specific Data Augmentation module. This module is strategically designed to further optimize contrastive learning for gaze-aware representations, generating two essential types of image pairs: gaze-consistent pairs and gaze-contrastive pairs.

- **Gaze-consistent pairs:** These are different views from the same full-face image, retaining gaze-related semantic features post-augmentation.
- **Gaze-contrastive pairs:** These are from two different images of the same subject, but with contrasting gaze features.

The generation process begins with a gaze-specific augmentation operator, termed *gaze-cropping*, which preserves essential gaze-related features:

1. **Facial landmark detection:** Identifies the eyes within the image.
2. **Bounding boxes creation:** Forms bounding boxes around the periocular areas based on the detected landmarks.
3. **Random cropping:** Ensures at least one periocular area is included in the cropped image.
4. **Resizing:** Resizes the cropped image to the original image's dimensions.

Color distortion is integrated as a secondary augmentation step to enhance the contrastive learning process. The GDA results from the sequential application of gaze-cropping and color distortion, producing gaze-consistent pairs.

Formally, the augmentation is modeled as a random variable T, and for a given subject i and image j, the image is denoted as $X_{i,j}$. The augmentation process involves randomly sampling two operators from T, denoted as $\{p, q\} \sim T$. A gaze-consistent pair $(X^p_{i,j}, X^q_{i,j})$ is generated by applying these operators to $X_{i,j}$. For gaze-contrastive pairs, $(X^p_{i,j}, X^q_{i,k})$ is constructed by applying p and q to different images $X_{i,j}$ and $X_{i,k}$ from the same subject. The paper[DZL23] includes examples of these pairs, demonstrating the GDA process.

4.5.2 SimCLR model with the gaze-specific data augmentation and subject-conditional projection module

In our study, we apply a modified approach to traditional contrastive learning, specifically tailored for gaze estimation tasks. The standard process involves a feature extractor $F(\cdot)$ mapping an augmented image X to a general representation space GP, extracting features $h = F(X) \in GP$. These features are then projected into an embedding space SP via a projection head $P(\cdot)$, creating embeddings $z = P(h) \in SP$. Typically, the projection head is a multi-layer perceptron (MLP) with one hidden layer, and the contrastive loss, often the InfoNCE loss, is applied in this embedding space SP.

However, in the context of gaze estimation, this method has limitations. The contrastive loss in SP and the inclusion of full-face images from diverse subjects in the mini-batch can cause $F(\cdot)$ to favor learning appearance or identity-related features. While these features distinguish between subjects, they are less effective in identifying gaze-aware features critical for gaze estimation.

To address these challenges, our research incorporates a novel strategy involving subject-specific contrastive losses[DZL23]. This is facilitated by using a subject-conditional projection module combined with a shared feature extractor. The objective is to direct the learning process more towards gaze-aware features, thereby enhancing the effectiveness and precision of gaze estimation models.

Subject-conditional Projection[DZL23]

In this study, we depart from conventional contrastive learning, which maps images to a uniform embedding space, by introducing a subject-conditional projection mechanism, $S(\cdot)$, creating individual embedding spaces, SP_i, for each subject. Our technique allows for subject-specific embeddings that capture distinctive features within their spaces.

The feature extractor $F(\cdot)$ works within these spaces to increase similarity for images with consistent gaze and decrease it for those with contrasting gaze directions from the same subject. This is aimed at enhancing the learning of gaze-specific features by $F(\cdot)$.

For each subject's image $X_{i,j}^t$, we one-hot encode the subject's identity ID_i, combine it with the features $h_{i,j}^t = F(X_{i,j}^t)$, and input this into an MLP to produce the embeddings $z_{i,j}^t = S(h_{i,j}^t)$.

4.6 Experiments

4.6.1 Experiment Objectives and Baseline

The primary goal of our experiments is to assess the performance of contrastive learning methods in gaze estimation tasks. This section introduces the various models used in our experimental setup:

1. **Original Model[ZSFB17a]**: Officially the original supervised learning model, proposed by Zhang et al., In this project, I have retrained and tested this model on the GazeCapture dataset. The result serves as our baseline for gaze estimation. It is used as a comparative benchmark against subsequent unsupervised contrastive learning models and is referred to as the Original Model throughout this paper.
2. **RanNet[DZL23]**: Comprising a feature extractor and a gaze estimator, both components of this model are initialized with random parameters and are not pre-trained.As a model with entirely random parameters, I have retrained and tested this model on the GazeCapture dataset. The result serves as a baseline for gaze estimation.
3. **SimCLR[CKNH20]**: This is the original SimCLR model combined with a gaze estimator head. The classic SimCLR contrastive learning framework is employed for pre-training and subsequently utilized for feature extraction, followed by a gaze estimator head for downstream gaze direction estimation tasks. In this document, it is abbreviated as the SimCLR model.

4. **ConEye[DZL23]**: Known as the SimCLR model with gaze-specific data augmentation and a gaze estimator head. The distinction from the SimCLR model is the use of gaze-specific data augmentation during the pre-training phase. The same gaze estimator head structure is employed for downstream gaze direction estimation tasks, and is referred to as ConEye in this study.
5. **ConGaze[DZL23]**: This model is an extension of ConEye, incorporating a subject-conditional projection module into the SimCLR model with gaze-specific data augmentation. It aims to project each subject's representations into their respective subspaces.

Using these five models, we conducted a series of experiments with datasets of various sizes and different parameter freezing strategies. The objective was to thoroughly explore the effectiveness of contrastive learning methods in gaze estimation tasks.

While the methods of these five models draw upon previous work in gaze estimation, in this study we have reconstructed them and conducted training and testing on the GazeCapture dataset. This preparation is key for the subsequent application of these models to eye contact detection tasks.

Baseline

Exp.	Model	Train Dataset	% Train Data Used	Test Dataset	MAE (°)
A	Orig. Model	GC	100%	GC_test	5.02
B	Orig. Model	MPII	100%	GC_test	7.63

Table 4.1: Baseline Performance. GC: GazeCapture, MPII: MPIIFaceGaze.

As demonstrated in Table 4.1, we initially trained the Original Model using the complete GazeCapture and MPIIFaceGaze training datasets, and both datasets include 3D gaze direction labels. The model was then evaluated using the test set of GazeCapture, achieving an optimal Mean Angle Error of 5.02°. This performance is established as the baseline standard for our study.

In addition, the hyperparameter configuration for training is: 20 epochs, 16 batch size, 0.1 base learning rate, 0.1 learning rate decay, SGD optimiser.

Pre-training

As illustrated in Table 4.2, we conducted pre-training on the feature extractors within the SimCLR, ConEye, and ConGaze models using the complete, unlabeled GazeCapture dataset. The metric of top-1 accuracy serves as an indicator of the quality of this pre-training. Specifically, it reflects each model's capability to distinguish between positive

Exp.	Model	Train Dataset	% Train Data Used	Test Dataset	Top-1 Acc. (%)
Pre-1	SimCLR FE	GC	100%	GC_test	93.75
Pre-2	ConEye FE	GC	100%	GC_test	98.44
Pre-3	ConGaze FE	GC	100%	GC_test	90.63

Table 4.2: Pre-training Performance FE: Feature Extractor of the Model

and negative samples, effectively measuring the discriminative power of the models in the pre-training phase.

In addition, the hyperparameter configuration for training was: 2 epochs, 32 batch size, 0.0003 learning rate, Adam optimiser.

Fine-tuning

In the subsequent phase of our study, we augmented the previously pre-trained feature extractors with a gaze estimator head, thus completing the assembly of the SimCLR, ConEye, and ConGaze models. We conducted five distinct experimental groups to fine-tune these models, each differing in the dataset used and the approach to the feature extractor parameters:

Group	**Models**	**Dataset**	**Extractor**
1	SimCLR, ConEye, ConGaze	1% GC	Frozen
2	SimCLR, ConEye, ConGaze, RanNet	1% GC	Unfrozen
3	SimCLR, ConEye, ConGaze	10% GC	Frozen
4	SimCLR, ConEye, ConGaze, RanNet	10% GC	Unfrozen
5	SimCLR, ConEye, ConGaze	100 samples GC	Frozen
6	RanNet	100 samples GC	Unfrozen

Table 4.3: Summary of Fine-Tuning Experiments

Table 4.3 provides a comprehensive overview of the six experimental groups. Group 1 involves fine-tuning the three models with 1% of the GazeCapture dataset, keeping the feature extractor parameters frozen. Group 2 expands this by including RanNet (as a baseline) and fine-tuning with unfrozen feature extractor parameters on the same subset. Groups 3 and 4 replicate this approach with 10% of the GazeCapture dataset, with Group 3 freezing and Group 4 unfreezing the feature extractor parameters. Finally, Group 5 employs a minimal dataset of 100 GazeCapture samples to fine-tune the models(SimCLR, ConEye, ConGaze) with frozen feature extractor parameters. In Experiment Group 6, we fine-tuned the RanNet model using only 100 samples from the GazeCapture dataset, with the feature extractor parameters left unfrozen.

Exp.	Model	Train Data	% Train Data Used	Test Data	MAE(°)
1	SimCLR (Froz.)	GC	1%	GC_test	9.71
2	ConEye (Froz.)	GC	1%	GC_test	8.57
3	ConGaze (Froz.)	GC	1%	GC_test	8.02

Table 4.4: Group 1 Fine-Tuning Results

Exp.	Model	Train Data	% Train Data Used	Test Data	MAE(°)
5	SimCLR (Unfroz.)	GC	1%	GC_test	6.23
6	ConEye (Unfroz.)	GC	1%	GC_test	5.46
7	ConGaze(Unfroz.)	GC	1%	GC_test	5.22

Table 4.5: Group 2 Fine-Tuning Results

In our study, we trained models on three distinct scales of GazeCapture data: 10%, 1%, and a subset of 100 samples. This strategy was designed to observe how the model's accuracy varies with the quantity of data. We aimed to determine if the best-performing model under the 10% data condition remains superior with only 100 samples, and to explore if there exists a model that maintains satisfactory accuracy while using less data.

The following six tables 4.4 4.5 4.6 4.7 4.8 4.9 provide detailed insights into the six experimental groups conducted in our study. Each table comprehensively outlines the specific parameters, methodologies, and outcomes associated with its respective experimental setup.

Training configuration: The six experimental groups conducted in this study were uniform in terms of hyperparameters. We employed a batch size of 16 and trained each model for 20 epochs. The L1 loss function was utilized as the primary loss metric. For optimization, the Stochastic Gradient Descent (SGD) method was implemented across all experiments.

Exp.	Model	Train Data	% Train Data Used	Test Data	MAE(°)
8	SimCLR (Froz.)	GC	10%	GC_test	9.15
9	ConEye (Froz.)	GC	10%	GC_test	8.04
10	ConGaze (Froz.)	GC	10%	GC_test	7.69

Table 4.6: Group 3 Fine-Tuning Results

Exp.	Model	Train Data	% Train Data Used	Test Data	MAE(°)
11	RanNet (Unfroz.)	GC	10%	GC_test	5.05
12	SimCLR (Unfroz.)	GC	10%	GC_test	5.07
13	ConEye (Unfroz.)	GC	10%	GC_test	4.76
14	ConGaze (Unfroz.)	GC	10%	GC_test	4.70

Table 4.7: Group 4 Fine-Tuning Results

Exp.	Model	Train Data	No. of Samples	Test Data	MAE(°)
15	SimCLR (Froz.)	GC	100 samples	GC_test	12.1
16	ConEye (Froz.)	GC	100 samples	GC_test	9.85
17	ConGaze (Froz.)	GC	100 samples	GC_test	8.12

Table 4.8: Group 5 Fine-Tuning Results

Exp.	Model	Train Data	No. of Samples	Test Data	MAE(°)
18	RanNet (Unfroz.)	GC	100 samples	GC_test	12.8

Table 4.9: Group 6 Fine-Tuning Results

5 Eye contact detection

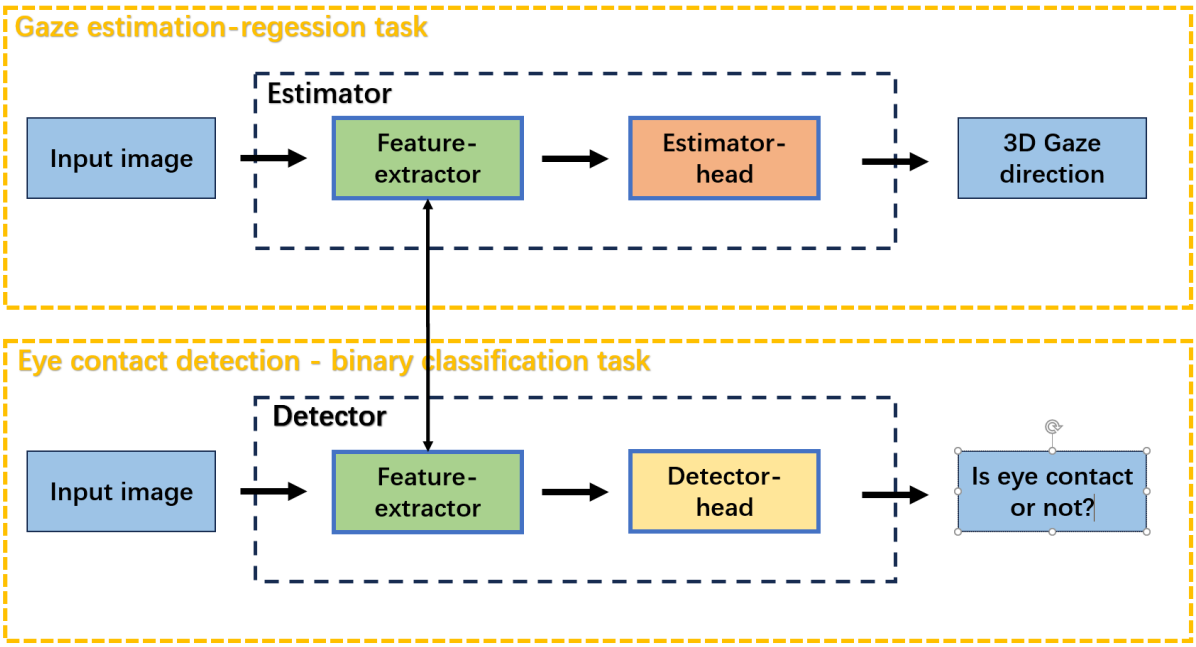

Figure 5.1: Relationship between gaze estimation and eye contact detection

Eye contact detection is a task intrinsically linked to gaze estimation, sharing a common foundation in feature extraction due to the analogous nature of the features required for both tasks. Unlike gaze estimation, which generates a vector to describe the 3D gaze direction, eye contact detection simplifies the output to a binary decision—determining the presence or absence of eye contact.

The shared feature extractor between the two tasks is central to this discussion. It indicates that while the tasks are different, the characteristics of the eyes necessary to infer gaze direction or eye contact are sufficiently similar to be captured by the same underlying neural network structures. However, this approach is not without limitations, as it relies on the feature extractor's ability to generalize well across both tasks.

Zhang et al. originally proposed a method in which a gaze estimation model[ZSB17], including its feature extractor, is first trained using supervised learning. The features learned are then repurposed for the eye contact detection task. Our paper posits that the

generality of the feature extractor, trained via supervised methods, can be enhanced. We propose training the feature extractor using contrastive learning to develop an extractor that is equally adept for both gaze estimation and eye contact detection.

There are two primary advantages to this method. First, it allows for the feature extractor to be trained without reliance on any labels, leveraging the self-supervised nature of contrastive learning. Second, a feature extractor that is more appropriately tuned to the shared aspects of both tasks can potentially increase the accuracy of the eye contact detection model, thereby delivering better performance across the board.

5.1 Datasets

5.1.1 EMVA Dataset

The EMVA dataset [ZSB17] is a unique collection designed to study mobile users' visual attention in everyday settings. It includes data from 32 participants, organized by individual IDs. Each participant's data is divided into "Recordings," which are further segmented into "Sessions" based on device activity. Sessions last up to 15 minutes and include a video file and other sensor data like acceleration and gyroscope, depending on the device's capabilities. This dataset provides an insightful look into real-world device usage patterns.

The EMVA dataset already includes a set of 15,740 images extracted from session videos of participants, these images have been artificially labelled with eye contact detection labels. My role involved shuffling these images and dividing them into a training set (70%) and a test set (30%). This split is essential for training and evaluating models that aim to understand and predict user attention in mobile environments.

5.2 Data Pre-processing

The preprocessing approach employed for eye contact detection adheres to the same protocol defined for gaze estimation to maintain uniformity. A comprehensive account of the preprocessing steps can be found in section 4.3 in Chapter 4.

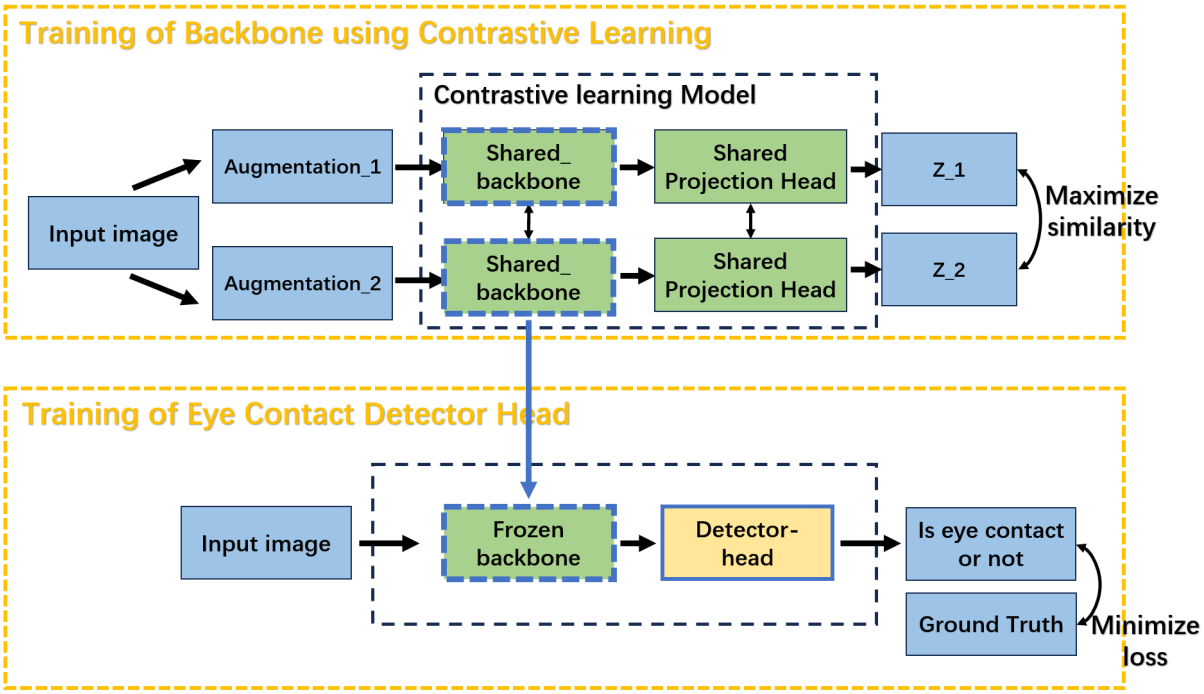

Figure 5.2: Contrastive Learning for Eye Contact Detection

5.3 Contrastive Learning for Eye Contact Detection

In the approach presented by Zhang et al.[ZSB17], a supervised learning model is trained for gaze estimation, incorporating a feature extractor whose extracted characteristics are subsequently shared with an eye contact detection model. This study suggests that the generalizability of the supervised feature extractor can be enhanced, proposing the utilization of contrastive learning for training a feature extractor that is equally effective for both tasks.

The key benefits of this approach include the training of the feature extractor without the need for labeled data and the potential for a more suitable feature extractor to increase the precision of the eye contact detection model.

As illustrated in Figure 5.2, after the initial feature extractor is trained using contrastive learning techniques, it is then frozen and utilized for the eye contact detection task. By leveraging the learned representations from the contrastive learning phase, where the backbone was trained to differentiate between augmentations of the same image, we now apply this robust feature extractor to classify the presence of eye contact, thus minimizing the classification loss. This shared approach between gaze estimation and eye contact detection tasks allows for a more generalized feature extraction process and aims to enhance the accuracy of eye contact classification.

5.4 Training Methods for Detection Head

In this section, we explore two distinct methodologies for training the detection head. The first approach is a supervised method that utilizes the EMVA training dataset. This method involves freezing the feature extractor and then training a Support Vector Machine (SVM) based detection head. It necessitates the EMVA training set annotated with eye contact labels.The selection of the EMVA dataset for our research is underpinned by its inclusion of eye contact labels. This key feature of the EMVA dataset aligns precisely with the objectives of our study, making it an ideal resource for our analysis in eye contact detection.

The second approach is unsupervised and allows for the use of the unlabelled EMVA dataset. Here, we utilize the most accurate available gaze estimator(the original methodology proposed by Zhang et al.[ZSFB17a]) to infer users' gaze points within the dataset. A clustering algorithm is then employed to determine which gaze points indicate eye contact. Labels generated via the original methodology proposed by Zhang et al.[ZSFB17a] are termed as pseudo-labels. These pseudo-labels enable us to train the detection head even in the absence of explicitly annotated eye contact data in the EMVA dataset.

5.4.1 Optics Clustering Algorithm

The Optics Clustering algorithm[ABKS99], employed in our unsupervised training method, plays a crucial role in categorizing gaze points for generating pseudo-labels. The Optics Clustering algorithm[ABKS99] is a method used in data analysis for identifying clusters in spatial data based on density.

In the context of our study, Optics Clustering is utilized to differentiate between gaze points that signify eye contact and those that do not. The algorithm iteratively adjusts cluster centroids and the assignment of data points to these clusters, aiming to minimize intra-cluster variance while maximizing inter-cluster distances. This iterative process continues until an optimal clustering solution is reached, based on predefined criteria or when convergence is observed.

The strength of Optics Clustering lies in its adaptability and efficiency in handling complex data structures, making it an ideal choice for processing the nuanced gaze data in the EMVA dataset.

5.5 Experiments

5.5.1 Experimental Series and Baseline Models

To assess the performance of SVM-based detection heads with feature extractors pre-trained through contrastive learning, we carried out a comprehensive set of experiments. We conducted four separate experimental series, each consisting of three trials to evaluate the three contrastive learning frameworks: SimCLR, ConEye, and ConGaze. The feature extractors were pre-trained using the full GazeCapture dataset. For the detection head training, the EMVA training set was employed, and evaluations were performed using the EMVA test set. The four experimental series were set up as follows:

1. In the first experimental series, the feature extractor pre-trained using contrastive learning was paired directly with an SVM detection head, which was then trained using pseudo-labels.
2. The second series used the same pre-trained feature extractor, but the SVM detection head was trained with ground truth labels.
3. The third experimental series involved an additional step of fine-tuning the pre-trained feature extractor using the configuration established in the fourth group of the gaze estimation experiments in table 4.7, prior to training the SVM detection head with pseudo-labels.
4. The fourth series mirrored the third, but here the SVM detection head was trained using ground truth labels.

Baseline models were also established for comparison, utilizing feature extractors trained with supervised methods on the GazeCapture dataset:

1. Baseline A: SVM detection head trained with ground truth labels.
2. Baseline B: SVM detection head trained with pseudo-labels.

These experimental and baseline setups aimed to evaluate the transferability of the contrastively learned features to the eye contact detection task and to explore the effectiveness of pseudo-labels for supervised training. The training and testing configurations for each setup are summarized in the table below.

CL: Contrastive Learning. **CL+FT**: Contrastive Learning with additional Fine-Tuning. **SVM**: Support Vector Machine, a machine learning model. **GT**: Ground Truth, referring to actual label data. **Pseudo**: Pseudo-Labels, labels generated by the original

Series	Extractor Training Method	Detection Head	Dataset for Detection Head	Evaluation Set
1	CL	SVM	EMVA Train, Pseudo	EMVA Test
2	CL	SVM	EMVA Train, GT	EMVA Test
3	CL+FT	SVM	EMVA Train, Pseudo	EMVA Test
4	CL+FT	SVM	EMVA Train, GT	EMVA Test
Baselines				
A	Supervised	SVM	EMVA Train,GT	EMVA Test
B	Supervised	SVM	EMVA Train,Pseudo	EMVA Test

Table 5.1: Summary of Experimental and Baseline Model Setups

supervised gaze estimation model proposed by Zhang et al.[ZSFB17a] and Optics Clustering[ABKS99]. **EMVA Train**: Training set from the EMVA dataset. **EMVA Test**: Test set from the EMVA dataset.

5.5.2 Overview of Experimental Series and Configurations

The results of the experiments for each series and for the baseline are shown in detail next in this section.

Table 5.2: Baseline Model Setups for Eye Contact Detection

Exp.	Feature Ectractor Framework	Dataset for Detection Head	Test Data	MCC
A	Supervised	EMVA Train, GT	EMVA Test	0.57
B	Supervised	EMVA Train, Pseudo	EMVA Test	0.46

Table 5.3: Series 1 Experimental Results for Eye Contact Detection

Exp.	Feature Ectractor Framework	Feature Ectractor Fine tuning	Dataset for Detection Head	Test Data	MCC
1	SimCLR	No	EMVA Train, Pseudo	EMVA Test	0.48
2	ConEye	No	EMVA Train, Pseudo	EMVA Test	0.57
3	ConGaze	No	EMVA Train, Pseudo	EMVA Test	0.63

Table 5.4: Series 2 Experimental Results for Eye Contact Detection

Exp.	Feature Ectractor Framework	Feature Ectractor Fine tuning	Dataset for Detection Head	Test Data	MCC
4	SimCLR	No	EMVA Train, GT	EMVA Test	0.48
5	ConEye	No	EMVA Train, GT	EMVA Test	0.63
6	ConGaze	No	EMVA Train, GT	EMVA Test	0.64

Table 5.5: Series 3 Experimental Results for Eye Contact Detection

Exp.	Feature Ectractor Framework	Feature Ectractor Fine tuning	Dataset for Detection Head	Test Data	MCC
7	SimCLR	Yes	EMVA Train, Pseudo	EMVA Test	0.41
8	ConEye	Yes	EMVA Train, Pseudo	EMVA Test	0.46
9	ConGaze	Yes	EMVA Train, Pseudo	EMVA Test	0.45

Table 5.6: Series 4 Experimental Results for Eye Contact Detection

Exp.	Feature Ectractor Framework	Feature Ectractor Fine tuning	Dataset for Detection Head	Test Data	MCC
10	SimCLR	Yes	EMVA Train, GT	EMVA Test	0.45
11	ConEye	Yes	EMVA Train, GT	EMVA Test	0.47
12	ConGaze	Yes	EMVA Train, GT	EMVA Test	0.48

6 Discussion

6.1 Comparative Analysis of Contrastive Learning Frameworks in Gaze Estimation

To explore the effectiveness of the three contrastive learning frameworks SimCLR, ConEye, and ConGaze in gaze estimation tasks, we conducted a total of 18 experiments, divided into 6 groups. The specific details of each experiment are outlined in the following tables: Table 4.4, Table 4.5, Table 4.6, Table 4.7, Table 4.8, and Table 4.9. In the following sections, we will select specific data points for comparison and discuss and analyze these findings in detail.

Group	**Models**	**Dataset**	**Extractor**
1	SimCLR, ConEye, ConGaze	1% GC	Frozen
2	SimCLR, ConEye, ConGaze, RanNet	1% GC	Unfrozen
3	SimCLR, ConEye, ConGaze	10% GC	Frozen
4	SimCLR, ConEye, ConGaze, RanNet	10% GC	Unfrozen
5	SimCLR, ConEye, ConGaze	100 samples GC	Frozen
6	RanNet	100 samples GC	Unfrozen

Table 6.1: Overview of Fine-Tuning Experiments

Exp.	Model	Train Data	Test Data	MAE(°)
A	Orig. Model	100% GC	GC_test	5.02
B	Orig. Model	100% MPII	GC_test	7.63

Table 6.2: Baseline

In our analysis, we observe notable differences within each group and across the two groups concerning the Mean Angle Error (MAE).

In Group 1 (Figure 6.1), where models underwent fine-tuning in a frozen state, ConGaze (Frozen) demonstrated the lowest MAE at 8.02°. This outperformance, relative to

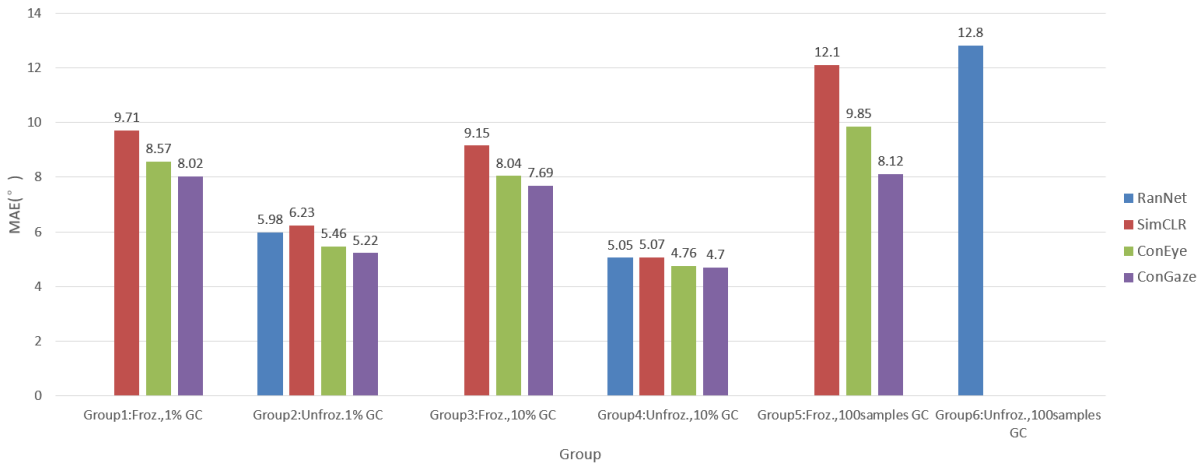

Figure 6.1: Overview of Gaze Estimation Results

ConEye (Frozen) and SimCLR (Frozen) with MAEs of 8.57° and 9.71° respectively, suggests ConGaze's feature extractor was more adept at identifying representative features for gaze estimation.

In Group 2 (Figure 6.1), involving unfrozen models, ConGaze (Unfrozen) continued to excel, registering the lowest MAE of 5.22°, followed by ConEye (Unfrozen) and SimCLR (Unfrozen) with MAEs of 5.46° and 6.23°, respectively. This superior performance trend of ConGaze is attributed to its feature extractor's ability to identify more representative features. Using this extractor as an initial state for fine-tuning resulted in better outcomes.

Comparing the two groups, the performance improvement in Group 2 is evident. For instance, SimCLR experienced a reduction in MAE from 9.71° in Group 1 to 6.23° in Group 2, while ConEye's MAE decreased from 8.57° to 5.46°, and ConGaze's MAE dropped from 8.02° to 5.22°. Overall, unfrozen models undergoing fine-tuning consistently yielded better results than their frozen counterparts, underscoring the effectiveness of allowing feature extractors to adjust and fine-tune for enhanced performance in gaze estimation tasks.

The results not only confirm unfrozen models' superiority in reducing MAE but also underscore the consistent efficacy of the ConGaze framework in both frozen and unfrozen states, surpassing SimCLR and ConEye in gaze estimation accuracy. It's important to note that models with frozen feature extractors do not alter these extractors' parameters during supervised fine-tuning. Consequently, unfrozen feature extractors post-fine-tuning can extract features more suitable for gaze estimation tasks. Supervised fine-tuning of the feature extractor tailors it more closely to the specific demands of the task.

Groups 3 and 4 (Figure 6.1 and 6.1) presented results that were largely consistent with those observed in Groups 1 and 2. However, a key distinction lies in the amount of training data used. While Groups 1 and 2 utilized only 1% of the GazeCapture dataset, Groups 3 and 4 employed 10%, leading to overall better results in the latter.

In Group 3, which involved fine-tuning with frozen models, ConGaze (Frozen) achieved the lowest MAE at 7.69°, followed by ConEye (Frozen) and SimCLR (Frozen) with MAEs of 8.04° and 9.15°, respectively. This trend aligns with the previous groups, where ConGaze consistently outperformed the others.

More notably, in Group 4, where models were fine-tuned without freezing, both ConGaze (Unfrozen) and ConEye (Unfrozen) surpassed the baseline model's performance. The baseline, trained with 100% of the labeled GazeCapture dataset, had an MAE of 5.02°. In contrast, ConGaze (Unfrozen) and ConEye (Unfrozen) in Group 4, trained with only 10% of the labeled dataset, achieved MAEs of 4.70° and 4.76°, respectively. To quantify, ConGaze (Unfrozen) exceeded the baseline by approximately 6.37%, and ConEye (Unfrozen) exceeded it by approximately 5.18%.

These results are particularly significant as they demonstrate that even with a reduced amount of training data, the fine-tuned ConGaze and ConEye models not only performed comparably but actually outperformed a model trained on the full dataset. This underscores the efficiency and effectiveness of these contrastive learning frameworks in gaze estimation tasks.This success is attributed to the gaze-specific data augmentation and subject-conditional projection module enhancements in the SimCLR framework, enabling the feature extractor to develop more expressive features for gaze estimation. Future research could involve fine-tuning ConGaze and ConEye with 100% of the GazeCapture dataset to potentially achieve even better model performance.

Analyzing the results from Groups 5 and 6, as outlined in Figure 6.1, offers valuable insights into the scenario where we have a significant amount of unlabeled data, but only a very limited set of labeled data is available for gaze estimation tasks.

In Group 5, where each model underwent fine-tuning with only 100 labeled samples from the GazeCapture dataset, ConGaze (Frozen) continued to show relatively strong performance with an MAE of 8.12°. However, the overall increase in MAE values across all models in this group, including ConEye (Frozen) and SimCLR (Frozen) with MAEs of 9.85° and 12.1°, respectively, underscores a key limitation: even with a feature extractor trained through contrastive learning, the scarcity of labeled data for fine-tuning still hampers the performance. This indicates that while unsupervised pre-training can extract useful features, the precision in tasks like gaze estimation still heavily relies on the availability of labeled data.

Group 6, with a single experiment involving RanNet (Unfrozen) and yielding an MAE of 12.8°, further reinforces this point. The lack of a contrastive learning-based pre-trained

feature extractor, combined with the limited labeled data, led to a more pronounced decrease in performance. This comparison distinctly shows that while unsupervised contrastive learning can alleviate the dependency on extensive labeled datasets, it cannot entirely replace the need for labeled data in fine-tuning for specific tasks. The stark difference in performance between Groups 5 and 6 highlights the value of using contrastive learning when labeled data is scarce but also emphasizes the necessity of some amount of labeled data to achieve optimal results in downstream tasks.

6.2 Comparative Analysis of Contrastive Learning Frameworks in Eye Contact Detection

The previous section analyzed the performance of feature extractors trained using contrastive learning frameworks in gaze estimation tasks, demonstrating their effectiveness due to the extraction of features pertinent to gaze estimation. This chapter extends this analysis to evaluate their performance in eye contact detection tasks, with a focus on three primary objectives.

Firstly, while there is an inherent correlation between eye contact detection and gaze estimation, and features suitable for gaze estimation may also be applicable to eye contact detection to some extent, it is imperative to empirically validate whether the features extracted by the contrastive learning frame trained extractors are equally suitable for eye contact detection. This necessitates a thorough examination of experimental data to confirm their applicability.

Secondly, we consider the impact of supervised fine-tuning of these feature extractors using the gaze-annotated GazeCapture dataset. Specifically, we aim to investigate how such fine-tuning influences the adaptability of the feature extractors to eye contact detection tasks.

Thirdly, the possibility of employing pseudo-labels is explored as a means to eliminate the dependence of the eye contact detection model on manually annotated eye contact labels. The effectiveness of using pseudo-labels will be scrutinized through experimental data, to ascertain whether this approach can maintain or enhance the performance of the detection model without relying on human-generated labels.

Through four series of experiments, this section seeks to assess and validate the versatility and effectiveness of feature extractors trained via contrastive learning frameworks in the domain of eye contact detection, exploring the potential benefits and limitations of this approach.

Series	Extractor Training Method	Detection Head	Dataset for Detection Head	Evaluation Set
1	CL	SVM	EMVA Train, Pseudo	EMVA Test
2	CL	SVM	EMVA Train, GT	EMVA Test
3	CL+FT	SVM	EMVA Train, Pseudo	EMVA Test
4	CL+FT	SVM	EMVA Train, GT	EMVA Test
Baselines				
A	Supervised	SVM	EMVA Train,GT	EMVA Test
B	Supervised	SVM	EMVA Train,Pseudo	EMVA Test

Table 6.3: Summary of Experimental and Baseline Model Setups for Eye Contact Detection

The configuration of the four series of experiments conducted in this study is presented in Table 6.3. In this context, 'CL' represents the three contrastive learning frameworks: SimCLR, ConEye, and ConGaze. Each series, along with the baseline experiments, has been designed to rigorously assess the performance of feature extractors trained using contrastive learning frameworks in the context of eye contact detection. The detailed results of these experiments are comprehensively documented in the following tables:

- Baseline model setups for Eye Contact Detection: Table 5.2.
- Results of Series 1 Experiments: Table 5.3.
- Results of Series 2 Experiments: Table 5.4.
- Results of Series 3 Experiments: Table 5.5.
- Results of Series 4 Experiments: Table 5.6.

These tables collectively provide an in-depth overview of the experimental outcomes, enabling a nuanced understanding of the efficacy of the applied contrastive learning methodologies in the domain of eye contact detection.

In comparing the first series of experiments, we observe distinct performances of ConGaze, SimCLR, and ConEye in eye contact detection tasks. ConGaze leads with an MCC score of 0.63, demonstrating superior performance, while SimCLR and ConEye achieve MCC scores of 0.48 and 0.57, respectively. ConGaze's outstanding performance is largely attributed to its integration of gaze-specific data augmentation and subject-conditional projection modules. These optimizations within the SimCLR framework enable the ConGaze feature extractor to develop features that are not only effective for gaze estimation but also highly suitable for eye contact detection tasks. Comparing the

Exp.	Feature Ectractor Framework	Dataset for Detection Head	Test Data	MAE
A	Supervised	EMVA Train, GT	EMVA Test	0.57
B	Supervised	EMVA Train, Pseudo	EMVA Test	0.46

Table 6.4: Baseline Results for Eye Contact Detection

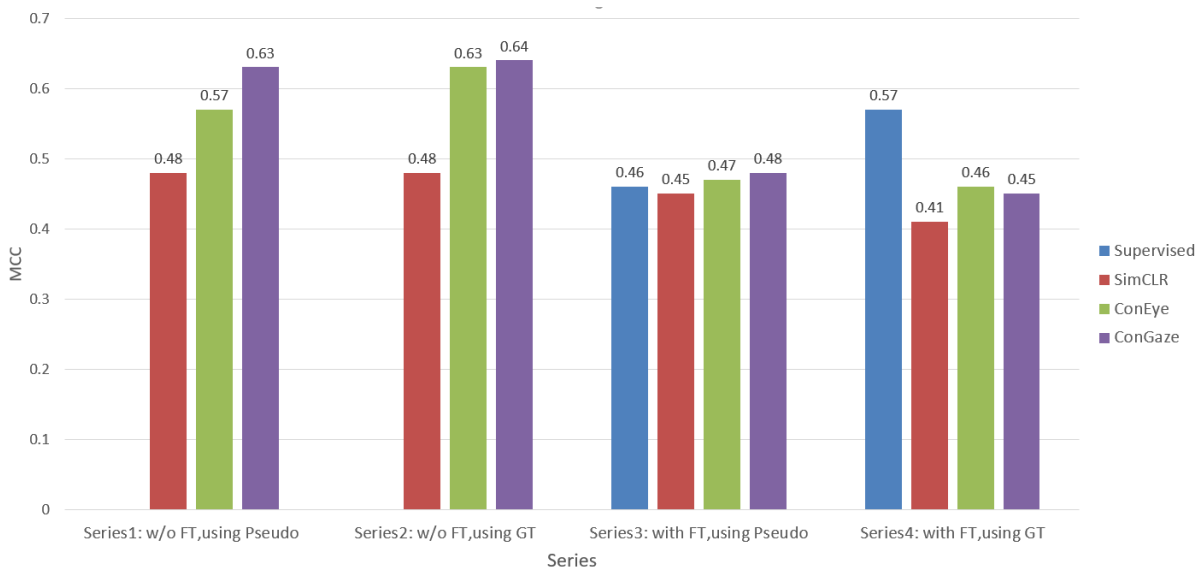

Figure 6.2: Overview of Eye Contact Detection Results

results, ConGaze not only outperforms SimCLR and ConEye with its higher MCC score but also surpasses the baseline standard. This indicates that the feature extractor in ConGaze is more effective than baseline models trained with supervised methods and labeled gaze estimation datasets. This highlights the advanced capability of ConGaze in extracting features optimal for eye contact detection tasks.

Cross-examining the results from Series 1 and Series 3 experiments, there is a discernible drop in the models' performance on eye contact detection after feature extractors were fine-tuned with gaze direction-labeled GazeCapture data. Specifically, SimCLR's MCC dropped from 0.48 to 0.41, ConEye's from 0.57 to 0.46, and ConGaze's from 0.63 to 0.45. This suggests that such fine-tuning made the extractors more specialized for specific gaze estimation, reducing their generalizability and effectiveness for eye contact detection. This specialization appears to compromise the versatility of the features, impacting their utility in broader tasks.

Comparing the results from Series 1 and Series 2, ConGaze maintains similar performance, achieving MCC scores of 0.63 and 0.64 respectively, whether trained with pseudo-labels or ground truth data. This consistency suggests the viability of using pseudo-labels for fine-tuning eye contact detection models. Conversely, ConEye shows a significant improvement when trained with ground truth labels, increasing from 0.57 to 0.63 MCC. This indicates that merely using gaze-specific data augmentation is insufficient to reduce dependency on ground truth labels for fine-tuning. Both gaze-specific data augmentation and subject-conditional projection modules are necessary to mitigate the impact of pseudo-labels' lower accuracy.

6.2.1 Limitations and Future Work

In this study, we focused on optimizing the SimCLR model for gaze estimation and eye contact detection tasks, but other contrastive learning models like GazeCLR[JM22] were not explored. Future research could build on this work by applying different contrastive learning models to eye contact detection. Additionally, after pre-training the feature extractor with contrastive learning, further fine-tuning with comprehensive datasets(100% GazeCapture) could enhance model performance.

A limitation of our evaluation phase was the exclusive use of GazeCapture, MPIIFaceGaze, and EMVA datasets due to time constraints. Future studies should incorporate cross-validation on more datasets and employ statistical methods to minimize errors.

Finally, the eye contact detection task lacks a dedicated dataset. The EMVA dataset's eye contact labels, manually added, may contain inaccuracies. Future efforts should focus on designing datasets with precise labeling during the data collection phase to ensure label accuracy.

7 Conclusion

Eye contact detection, intricately linked with gaze estimation, shares a foundational basis in feature extraction due to the similar nature of required features for both tasks. However, unlike gaze estimation which computes a vector for 3D gaze direction, eye contact detection simplifies to a binary output, determining the presence or absence of eye contact. This thesis addresses a notable gap: the scarcity of dedicated datasets for eye contact detection and the challenge in generalizability due to differing labels between these two tasks.

While gaze estimation has been extensively researched with a variety of methods and datasets, eye contact detection has not received equivalent attention. This is partly due to the difficulty in directly applying gaze estimation techniques for inferring eye contact, often resulting in lower accuracy. Zhang et al. initially proposed a method[ZSB17] where a gaze estimation model, including its feature extractor, is trained using supervised learning, and the features learned are then repurposed for eye contact detection. However, this method's limitations stem from the reliance on the feature extractor's ability to generalize across both tasks.

In this thesis, we propose a novel model construction for eye contact detection that employs an unsupervised contrastive learning approach. This methodology allows the utilization of extensive gaze estimation datasets to train the feature extractor for eye contact detection, followed by fine-tuning with a smaller, labeled eye contact dataset. This approach is groundbreaking in applying contrastive learning to eye contact detection.

Our study applied the SimCLR contrastive learning model, specifically optimized for eye contact detection, resulting in significant accuracy improvements. This method's dual advantages include training the feature extractor without relying on labels and tuning it to better suit the shared aspects of both gaze estimation and eye contact detection tasks.

In summary, this thesis demonstrates that contrastive learning can train feature extractors with enhanced generalizability, capable of extracting features apt for both gaze estimation and eye contact detection tasks, thereby enhancing the overall performance in both domains.

Bibliography

[ABKS99] M. Ankerst, M. M. Breunig, H.-P. Kriegel, J. Sander. "OPTICS: Ordering Points To Identify the Clustering Structure." In: *ACM SIGMOD Record*. Vol. 28. 2. ACM. 1999, pp. 49–60 (cit. on pp. 35, 56, 58).

[BGL+94] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah. "Siamese neural networks for one-shot image recognition." In: *Proceedings of the 1994 IEEE International Conference on Neural Networks* 1 (1994), pp. 269–274 (cit. on p. 25).

[CKNH20] T. Chen, S. Kornblith, M. Norouzi, G. Hinton. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. arXiv: 2002.05709 [cs.LG] (cit. on pp. 26, 27, 47).

[Dee23] DeepInsight. *InsightFace GitHub Repository*. Accessed: December 7, 2023. 2023. URL: https://github.com/deepinsight/insightface (cit. on p. 40).

[DZL23] L. Du, X. Zhang, G. Lan. *Unsupervised Gaze-aware Contrastive Learning with Subject-specific Condition*. 2023. arXiv: 2309.04506 [cs.CV] (cit. on pp. 44–48).

[Hin02] G. E. Hinton. "Training products of experts by minimizing contrastive divergence." In: *Neural computation* 14.8 (2002), pp. 1771–1800 (cit. on p. 25).

[HZRS16] K. He, X. Zhang, S. Ren, J. Sun. "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 21–23).

[JM22] S. Jindal, R. Manduchi. *Contrastive Representation Learning for Gaze Estimation*. 2022. arXiv: 2210.13404 [cs.CV] (cit. on pp. 32, 33, 67).

[KKK+16] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, A. Torralba. "Eye tracking for everyone." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2176–2184 (cit. on p. 38).

[LBH15] Y. LeCun, Y. Bengio, G. Hinton. "Deep learning." In: *Nature* 521.7553 (2015), pp. 436–444 (cit. on pp. 15, 18).

[LIAP22] F. Llorella, E. Iáñez, J. Azorin, G. Patow. “Classify four imagined objects with EEG signals.” In: *Evolutionary Intelligence* 15 (Sept. 2022). DOI: 10.1007/s12065-021-00577-y (cit. on p. 19).

[RFB15] O. Ronneberger, P. Fischer, T. Brox. “U-net: Convolutional networks for biomedical image segmentation.” In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241 (cit. on p. 23).

[RHGS15] S. Ren, K. He, R. Girshick, J. Sun. “Faster R-CNN: Towards real-time object detection with region proposal networks.” In: *Advances in neural information processing systems*. 2015, pp. 91–99 (cit. on p. 23).

[RKVT15] A. Recasens, A. Khosla, C. Vondrick, A. Torralba. “Where are they looking?” In: *Advances in Neural Information Processing Systems*. 2015, pp. 199–207 (cit. on p. 34).

[RVKT16] A. Recasens, C. Vondrick, A. Khosla, A. Torralba. “Following gaze across views.” In: *arXiv preprint arXiv:1612.03094* (2016) (cit. on p. 34).

[SKP15] F. Schroff, D. Kalenichenko, J. Philbin. “FaceNet: A unified embedding for face recognition and clustering.” In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823 (cit. on p. 25).

[SMS14] Y. Sugano, Y. Matsushita, Y. Sato. “Learning-by-synthesis for appearance-based 3d gaze estimation.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1821–1828 (cit. on pp. 39, 40).

[Soc17] A. M. Soccini. “Gaze estimation based on head movements in virtual reality applications using deep learning.” In: *2017 IEEE Virtual Reality (VR)*. 2017, pp. 413–414. DOI: 10.1109/VR.2017.7892352 (cit. on p. 37).

[SYFN13] B. A. Smith, Q. Yin, S. K. Feiner, S. K. Nayar. “Gaze locking: passive eye contact detection for human-object interaction.” In: *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM. 2013, pp. 271–280 (cit. on p. 34).

[YLL+15] Z. Ye, Y. Li, Y. Liu, C. Bridges, A. Rozga, J. M. Rehg. “Detecting bids for eye contact using a wearable camera.” In: *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference on*. IEEE. 2015 (cit. on p. 34).

[ZSB17] X. Zhang, Y. Sugano, A. Bulling. “Everyday eye contact detection using unsupervised gaze target discovery.” In: *Proceedings of the 30th annual ACM symposium on user interface software and technology*. 2017, pp. 193–203 (cit. on pp. 11, 34, 35, 53–55, 69).

[ZSB18] X. Zhang, Y. Sugano, A. Bulling. “Revisiting data normalization for appearance-based gaze estimation.” In: *Proceedings of the 2018 ACM symposium on eye tracking research & applications*. 2018, pp. 1–9 (cit. on pp. 39, 41).

[ZSFB17a] X. Zhang, Y. Sugano, M. Fritz, A. Bulling. “It’s Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation.” In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, July 2017. DOI: 10.1109/cvprw.2017.284. URL: http://dx.doi.org/10.1109/CVPRW.2017.284 (cit. on pp. 32, 38, 42, 47, 56, 58).

[ZSFB17b] X. Zhang, Y. Sugano, M. Fritz, A. Bulling. “Mpiigaze: Real-world dataset and deep appearance-based gaze estimation.” In: *IEEE transactions on pattern analysis and machine intelligence* 41.1 (2017), pp. 162–175 (cit. on p. 40).

All links were last followed on March 17, 2008.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature