PEDESTRIAN ATTENTION RECOGNITION USING CNN-BASED EYE DETECTION

Aaron Park
Student# 1010195101
aaron.park@mail.utoronto.ca

Axel Pena Hernandez
Student# 1011316096
axel.pena@mail.utoronto.ca

Jude Hasbini
Student# 1010066054
jude.hasbini@mail.utoronto.ca

Nadeem Bakr
Student# 1008997150
n.bakr@mail.utoronto.ca

ABSTRACT

This project explores pedestrian attention recognition using deep learning by classifying whether pedestrians are attentive or distracted. Leveraging the Gaze360 dataset, we trained a convolutional neural network (CNN) on grayscale and downsampled images to perform binary classification based on gaze & head pose alignment. Our preprocessing pipeline employs vector normalization, angle-based labeling, and class balancing to mitigate bias within the dataset. Evaluation across training, validation, test demonstrated strong generalization – with an accuracy of around 87%. Our model prioritized recall to reduce false negatives but left leeway for false positives. Qualitative analysis highlighted the model's limitations, such as its difficulty with classifying images with similar background and pedestrian colour. This model has real-world potential for enhancing pedestrian safety in both high and low-traffic urban environments.

1 INTRODUCTION

In 2021, crossing intersections was the action that led to the most pedestrian accidents in Ontario Government of Ontario. Distracted walking, which has surged as more people use headphones and smartphones in public, reduces situational awareness and increases the risk of accidents. Our project, Pedestrian Attention Recognition, aims to improve road safety by detecting pedestrian attentiveness in real time and issuing a warning when a pedestrian is classified as distracted. CNNs will be used to analyze images of pedestrians' faces, use feature maps to identify face patterns and eye movements and then apply fully connected layers to classify whether the pedestrian is attentive or distracted. This is something simple computer vision such as rule based systems struggles with without manual tuning.

2 ILLUSTRATION

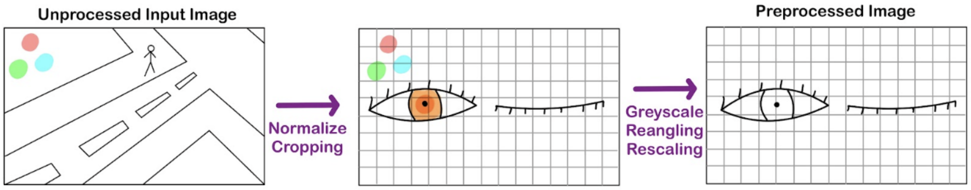

Figure 1: Timeline for preprocessing data.

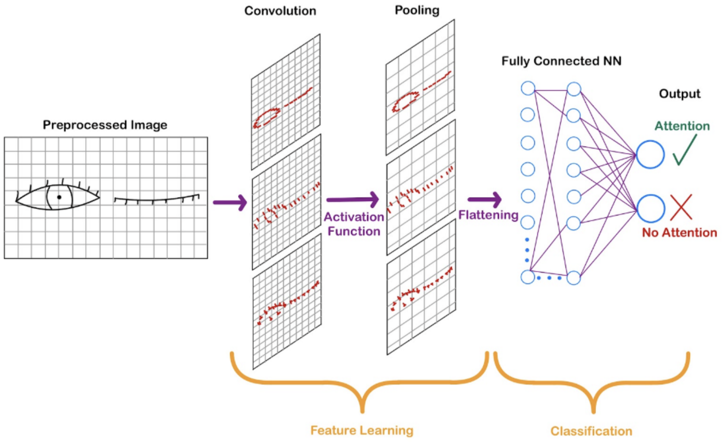

Figure 2: Timeline for the data through our CNN model.

3 Background & Related Work

3.1 MPIIGaze

The MPIIGaze dataset et al. (e) comprises 213,659 full face images from 15 participants captured via laptop webcams during daily laptop use. Their approach used a convolutional neural network (CNN) trained to estimate gaze direction using RGB images under unconstrained, real-world conditions, including varying head poses, lighting, and environments. The dataset is annotated with ground truth 2D gaze vectors (yaw and pitch) and 3D head pose information, making it especially useful for modeling attention-related behaviors.

This work is foundational to our project, which seeks to detect whether pedestrians are attentive or distracted in real-world scenes. While their model focuses on estimating the precise gaze vector, our system builds upon this by interpreting gaze–head pose alignment to classify attentiveness. Their attention to environmental variability makes the dataset and techniques highly transferable to our context.

3.2 iTracker

The GazeCapture dataset et al. (d) comprises over 2.5 million frames from approximately 1,474 unique participants. Data was collected via a mobile app on iPhones and iPads, making this the first large-scale dataset built through crowdsourced contributions in real-world settings. The model itself is a CNN that processes four inputs: left eye image, right eye image, full face image, and a face grid mask. These inputs were used to predict 2D gaze coordinates on mobile device screens. iTracker achieved accuracy under varied lighting conditions, head poses, and device types, without requiring specialized hardware or calibration.

Though iTracker's target domain was mobile gaze tracking, the environmental constraints it had are similar to those in pedestrian environments such as urban intersections. Moreover, iTracker demonstrated that a CNN could infer gaze direction effectively from facial features.

3.3 OpenFace

OpenFace et al. (b) is a toolkit for facial behavior analysis that uses facial landmark detection, head pose estimation, eye gaze estimation, and facial Action Unit recognition. Developed by Baltrušaitis et al., OpenFace was the first publicly available system to offer all these capabilities simultaneously using only standard RGB webcam input, without requiring specialized hardware. Its modular pipeline can process live or recorded video streams, delivering 2D and 3D facial landmarks, gaze vectors, and head orientation in real time.

OpenFace is particularly useful as a preprocessing tool for extracting face, eye, and head pose features from pedestrian video frames. This allows us to identify whether a subject is looking forward or away, which is critical for determining attentiveness.

3.4 Pedestrian Detection Methods

Dollár et al.'s work et al. (a) introduced one of the first benchmark frameworks for pedestrian detection by evaluating multiple algorithms including HOG, SVMs, and boosted decision trees. They released the Caltech Pedestrian Dataset, which contains approximately 250,000 video frames and over 350,000 annotated bounding boxes captured from a moving vehicle in urban environments. Their evaluation process allowed for consistent detection methods in pedestrian perception systems. This enables researchers to train, test, and compare pedestrian detection systems using a standardized dataset.

3.5 Gaze-Based Visual Alert Systems for Automotive Safety

Palinko et al. (c) conducted a controlled driving simulator study to investigate the use of eye tracking in detecting cognitive distraction during driving. Their research focused on measuring pupil diameter and gaze behavior to estimate driver mental workload, demonstrating that increased cognitive load leads to noticeable changes in pupil size, even under constant lighting. While the study did not implement a real-time alert system, it provided strong experimental evidence that eye-based metrics such as fixation duration, gaze shift frequency, and pupil response are effective indicators of distraction and cognitive strain in high-risk environments like driving. Although their work centered on drivers, gaze behavior can reflect attention and is directly applicable to our project.

4 Data Processing

For this project, we used the **Gaze360** dataset, which contains full head images along with precise 3D gaze direction vectors and head pose annotations. This dataset was better suited to our task than MPIIGaze because it provided a wider field of view and richer contextual information, allowing us to leverage cues such as head orientation, eye position within the head, and subtle posture differences. Our goal was to adapt Gaze360 for a **binary attention classification task**, labeling each image as either *attentive* or *distracted* based on the angular difference between the person's gaze direction and the camera's forward axis

4.1 Steps Taken

1. **Reading and Labeling Data**
   - Extracted **3D gaze direction vectors** from the dataset's metadata.
   - Normalized each gaze vector and compared it to the camera's forward axis.
   - Computed the angular distance (in degrees) between the gaze vector and forward direction.
   - Applied a **20°** threshold:
     - ○ $\leq 20^\circ \rightarrow$ Attentive (label = 1)
     - ○ $> 20^\circ \rightarrow$ Distracted (label = 0)
   - Saved all results (image paths, gaze angles, labels) to a CSV for downstream processing.

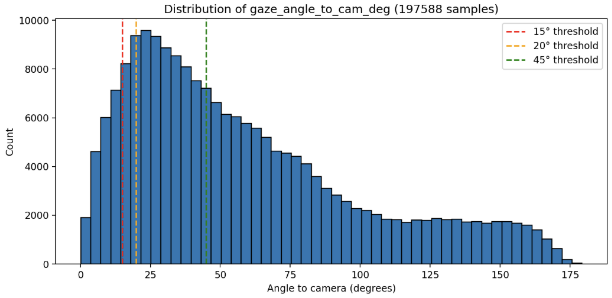

Figure 3: Distribution of Angle Differences.

2. **Class Imbalance Handling**
   - Initial counts: large imbalance, with many more distracted than attentive samples.
   - To prevent bias, we **downsampled** the majority class so both classes had an equal number of samples.

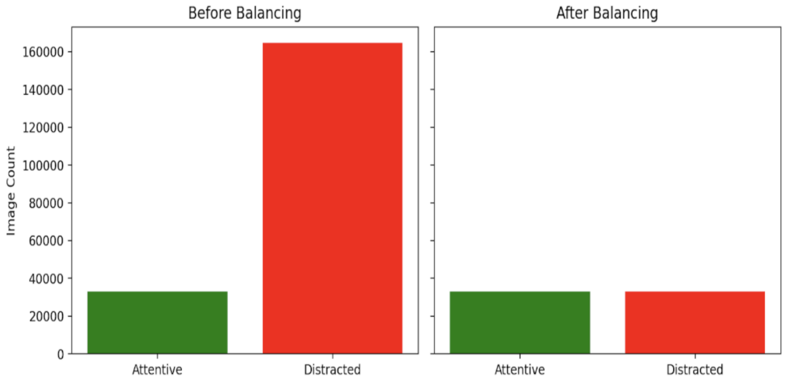

Figure 4: Before and After Balancing.

- This ensured balanced exposure to both classes during training.

3. **Preprocessing Images**
   - Converted all images to **grayscale** to reduce computational complexity while keeping key facial features.
   - Resized each image to **32×32 pixels**, balancing detail retention with faster training.
   - Saved images into separate folders: `/attentive/` and `/distracted/` for easy identification and model loading.
   - Created a final CSV index linking original paths to processed files.

4. **Processing Environment**

- Initial processing in Google Colab proved too slow for the full dataset (∼150k+ images).
- Moved processing to a local VS Code environment with optimized file I/O.

5 ARCHITECTURE

Our final model architecture consists of a convolutional neural network as the embedding for feature extraction, followed by two fully connected layers as the classifier. The expected input is a 32x32 grayscale image where the output is a binary classification. The model contains three convolutional layers, all with similar properties. 2-dimensional convolution is applied with a 3x3 filter. All layers have a stride and padding of one. This ensures a simple consistency. Following each convolutional layer, a ReLU activation function is applied.

Furthermore, max pooling is also applied following the ReLU function proceeding convolution layers one and two. A 2x2 filter is used for all pooling to reduce output size by a factor of two in each dimension. To ensure our model is able to learn deeper, our architecture consists of methods to mitigate a possible vanishing or exploding gradient. These include the addition of two auxiliary outputs and one skip connection.

Auxiliary output 1 branches out following the pooling layer of the second convolutional layer. Auxiliary output 2 branches out before the embedding is flattened, nearing the end of the forward pass.

In terms of the skip connection, an identity is stored after the first convolutional layer and ReLU, before max pooling. This identity directly connects to the same point that Auxiliary output 2 emerges from. That is, at the final stage of feature extraction, preceding flattening.

6 BASELINE MODEL

To establish a point of comparison before developing our more complex CNN model, we implemented a baseline using a Support Vector Machine (SVM) with a radial basis function (RBF) kernel.

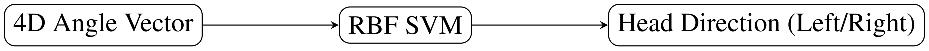

Figure 5: Baseline processing pipeline.

6.1 Input features and Task Formalisation

This baseline model utilizes a simple classification task: distinguishing whether a pedestrian's head is turned to the left or to the right, based on four angular features (vectors) extracted from each image. These features are `gaze_yaw`, `gaze_pitch`, `head_yaw`, and `head_pitch`.

6.2 Classifier

An RBF–SVM (`C=1`, `gamma=scale`) was trained on an 80/20 split of a class balanced subset (38 499 images per class); implementation was a single `scikit-learn` call. The classification's simplicity in conjunction with the use of low-dimensional input features resulted in extremely high performance, with training and testing accuracies being 99.51 and 99.36 %, respectively.

Table 1: Baseline SVM performance accuracy on training and test splits and confusion matrix on the test set.

	Train	Test
Accuracy	0.9951	0.9936

	Predicted	
True	Left	Right
Left	7602	98
Right	1	7699

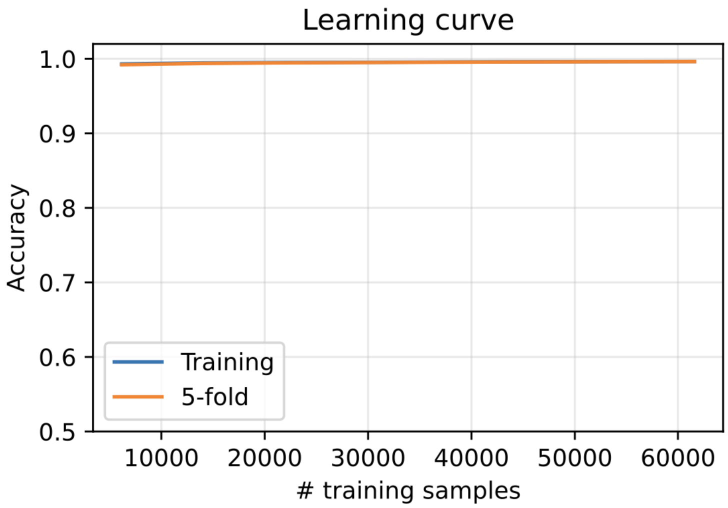

Figure 6: Baseline SVM learning curve.

Figure 4 shows the learning curve for our baseline SVM. Here, ”5-fold” refers to the average accuracy obtained from 5-fold cross validation, which provided an estimation on how well the model generalized to new data.

Although the SVM achieves next to perfect accuracy, it is important to note that this baseline model operates on a fundamentally different and simpler problem than our CNN. The CNN is tasked with learning to detect attentiveness directly from raw image data, which is a significantly harder problem as it operates on the more complex connection of where someone's head is facing and where their eyes are looking. Therefore, even though the baseline model yields very high accuracy, it relies on the much simpler task of distinguishing whether a person's head is facing left or right - highlighting the more complicated nature of the CNN.

6.3 CHALLENGES FACED

Initially, the baseline was originally designed to label data as either attentive or distracted based on a 45° threshold between head pose and gaze. However, this produced a heavily imbalanced dataset, which is why the current baseline model instead relies on head direction (left or right), which yielded a much more balanced and simpler problem for the SVM.

7 QUANTITATIVE RESULTS

To evaluate our model, we measured loss, error rate, accuracy, and recall across the training, validation, and test sets (Table 2). We prioritized **recall** over accuracy as in our application, a false negative (i.e., failing to detect a distracted pedestrian) was more dangerous than a false positive (i.e., incorrectly classifying an attentive pedestrian as ”distracted”), which is not as much of a real-world hazard.

Table 2: Model’s performance on training, validation, and test sets at epoch 18.

Dataset	**Loss**	**Error Rate**	**Accuracy**	**Recall**
Training	0.445	0.090	0.910	0.880
Validation	0.631	0.135	0.8652	0.854
Test	0.626	0.130	0.870	0.8646

As shown in Table 2, the training set achieved a loss of 44.5%, accuracy of 91.0%, and recall of 88.0%. The validation set achieved a loss of 63.1%, accuracy of 86.5%, and recall of 85.4%. The test set performed similarly, with a loss of 62.6%, accuracy of 87.0%, and recall of 86.46% – indicating strong generalization.

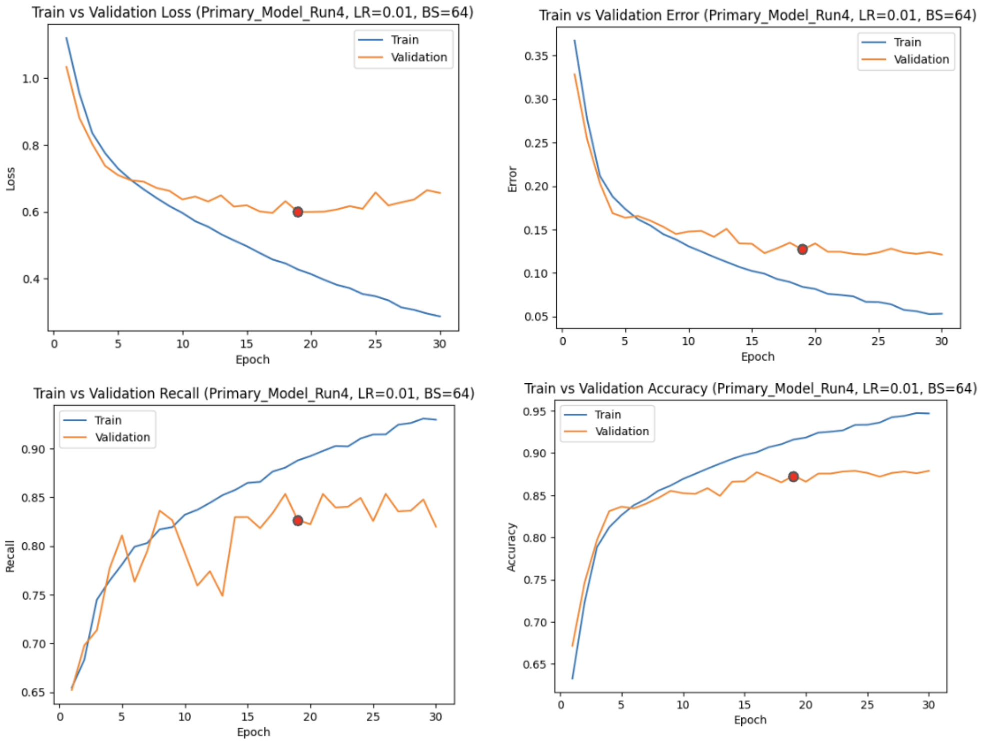

Figure 7: Training vs Validation Curves for loss, error, recall, and accuracy, respectively (left to right)

Figure 7 shows training vs. validation curves for loss, error, recall, and accuracy. Both loss curves gradually decrease; a small gap is present which indicates small overfitting. The validation accuracy and recall curves plateau around epoch 18, while the training curves continue increasing, which indicate that further training leads to only marginal improvement.

8 QUALITATIVE RESULTS

Sample Selection: Five images were randomly chosen from the held-out test set (Figure 8) to avoid reporting cherry-picked results (i.e., only choosing images that were correctly classified).

Performance Context: Our model was primarily selected for its high recall, which minimizes false negatives but has leeway for false positives, as mentioned previously.

Observations: In most cases, the model correctly identified whether a pedestrian is attentive or distracted. However as shown in the cherry-picked failure example, the model struggles when background regions have similar colouring to the pedestrian's features (e.g., The lady's hair blending in with the background in Figure 9).

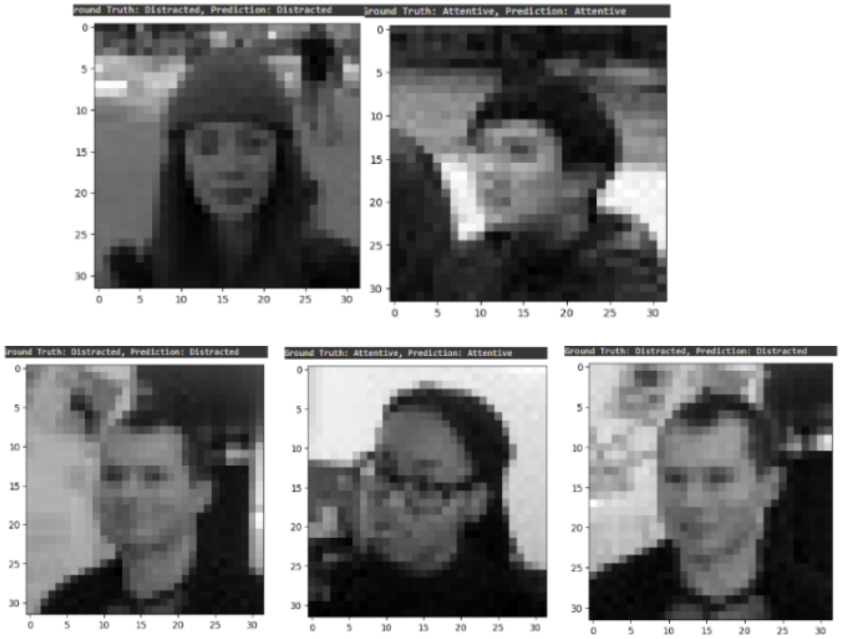
round Truth: Distracted, Prediction: Distracted
Ground Truth: Attentive, Prediction: Attentive
round Truth: Distracted, Prediction: Distracted
Ground Truth: Attentive, Prediction: Attentive
Ground Truth: Distracted, Prediction: Distracted
0
5
10
15
20
25
30

Figure 8: Model correctly outputs the right classification for all five images.

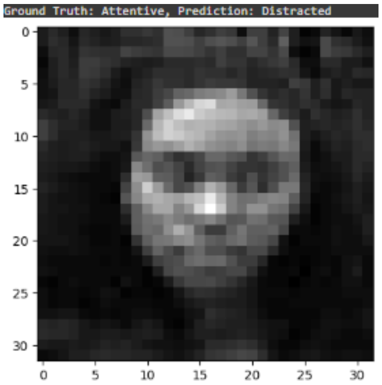

Figure 9: Model incorrectly outputs the classification due to similar background and pedestrian colouring.

Note: the images are separate inputs, and the label at the top of each image shows the ground truth and prediction.

9 EVALUATE MODEL ON NEW DATA

We used a 60% training, 10% validation, 10% training test split, and set aside 20% of the Gaze360 dataset as unseen data. This unseen data was not used for hyperparameter tuning or performance evaluation during model development, ensuring a fair and unbiased measure of generalization. While we were unable to source a completely independent dataset, reserving a portion of Gaze360 still provides a meaningful test, as these samples include different subjects, head poses, and environments from the training set. In practice, the model would process new data from a live intersection feed, and performance on the unseen data should closely reflect real-world results, assuming the same preprocessing steps are applied.

Our unseen dataset resulted in an accuracy of 0.8656 and a recall of 0.8563, which is sufficient for the intended application. Occasional false positives do not create harm, as they simply prompt additional awareness in pedestrians who are already attentive. The cost of a missed detection is therefore low compared to the potential safety benefits of correct detections. While this accuracy may appear modest in absolute terms, it should be interpreted in the context of the data and model constraints: our evaluation used low-resolution images, whereas real-world deployment could leverage higher-resolution cameras and more optimized preprocessing, likely improving performance. These results indicate the model is already viable for deployment, with potential for even stronger performance under real-world conditions.

10 DISCUSSION

Considering the achieved accuracy of 0.91 and 0.87 on train data and test data respectively, it is clear that our model is performing well. Through optimizing parameters, we have found that a learning rate of 0.01, batch size of 64, and 30 epochs is ideal considering the size of our dataset.

Due to the nature of our project, our model prioritises high recall to minimise false negatives. False positive outputs are more accepted than false negatives for safety purposes. It is safer in application for the model to gain a pedestrian's attention who is already focused, than to falsely predict that they are. Due to the variety of environments and lighting conditions present in our dataset, our model was able to learn deeper on key features to minimize false predictions. Because of this, it can be argued that the type of data that we used, incorporating complexity, contributed to our model's high accuracy scores.

Previously, we had attempted to train the model on inputs of size 128x128, however, this led to very long training times. Then, we reduced the input size to 64x64, yet our model was still too slow. It was not until we changed the size to be 32x32 that training times became acceptable. Due to the addition of auxiliary outputs, more computation must be performed which can lengthen training speeds. We found that in many aspects of this project, there was a give and take dynamic where in this case, incorporating methods to tackle one problem (vanishing gradient) can lead to others (slower training speeds).

Despite the low error and loss of the model, there are still issues that must be addressed to ensure successful applications. A key observation is that the model faces difficulty accurately classifying an image where the individual is similar in color to their background. This is understandable especially due to the fact that we train and test on grayscale images, meaning the model only has one input channel to extract from. By increasing image contrast in preprocessing, this issue may be resolved.

Overall, our model is performing as expected with minimal issues. At this stage, it is prepared for application. However, considerations on optimizing the model revolve around modifying preprocessing techniques and potentially adding more skip connections to minimize processing times.

11 ETHICAL CONSIDERATIONS

The Gaze360 dataset Kellnhofer et al. (2019) includes diverse environments, illumination levels, ages, ethnicities, head poses, and gaze directions, which helps mitigate representation bias in our model. However, for certain groups, such as age and ethnicity, the exact distribution is unknown, as samples were collected without explicit consideration of these sensitive attributes. This raises the possibility of evaluation bias for some demographic groups.

This is important because eye size can influence gaze estimation error, and eye size may correlate with certain demographics. Another limitation is that our model is intended for outdoor use at intersections, while the dataset contains both indoor and outdoor environments. This could introduce deployment bias, though it may also improve generalization.

Finally, there is an ethical concern regarding privacy, as deployment would require live recording of individuals. Whether the model stores new data for retraining could significantly affect user privacy.

REFERENCES

Dollar et al. Pedestrian detection: An evaluation of the state of the art. https://pdollar.github.io/files/papers/DollarCVPR09peds.pdf, a. Accessed: 2025-06-10.

Kim et al. Pva net: Lightweight deep neural networks for real-time object detection. https://ieeexplore.ieee.org/document/7477553, b. Accessed: 2025-06-10.

Kun et al. Behavioral patterns in eye gaze during distracted driving. https://andrewkun.com/papers/2012/etra2012_paper164_final.pdf, c. Accessed: 2025-06-10.

Zhang et al. Joint face detection and alignment using multi-task cascaded convolutional networks. https://arxiv.org/pdf/1606.05814, d. Accessed: 2025-06-10.

Zhang et al. Appearance-based gaze estimation in the wild. https://openaccess.thecvf.com/content_cvpr_2015/papers/Zhang_Appearance-Based_Gaze_Estimation_2015_CVPR_paper.pdf, e. Accessed: 2025-06-10.

Government of Ontario. Ontario road safety annual reports (orsar). https://www.ontario.ca/document/ontario-road-safety-annual-reports-orsar. Accessed: 2025-07-11.

Petr Kellnhofer, Adrià Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6912–6921, October 2019. URL https://gaze360.csail.mit.edu/. Accessed: 2025-08-15.