A Fully Automatic Visual Attention Estimation Support System for A Safer Driving Experience

Francesca Fiani[1], Samuele Russo[2] and Christian Napoli[1,3,4]

[1]Department of Computer, Control and Management Engineering, Sapienza University of Rome, 00185 Roma, Italy
[2]Department of Psychology, Sapienza University of Rome, 00185 Roma, Italy
[3]Institute for Systems Analysis and Computer Science, Italian National Research Council, 00185 Roma, Italy
[4]Department of Computational Intelligence, Czestochowa University of Technology, 42-201 Czestochowa, Poland

Abstract

Drivers' attention is a key element in safe driving and in avoiding possible accidents. In this paper, we present a new approach to the task of Visual Attention Estimation in drivers. The model we introduce consists of two branches, one which performs Gaze Point Detection to determine the exact point of focus of the driver, and the other which executes Object Detection to recognize all relevant elements on the road (e.g. vehicles, pedestrians, and traffic signs). The combination of the two outputs from the two branches allows us to determine whether the driver is attentive and, eventually, on which element of the road they are focusing. Two models are tested for the gaze detection task: the GazeCNN model and a model consisting of a CNN+Transformer. The performance of both models is evaluated and compared with other state-of-the-art models to choose the best approach for the task. Finally, the results of the Visual Attention Estimation performed on 3761 pairs of images (driver view and corresponding road view) from the DGAZE dataset are reported and analyzed.

Keywords

Visual Attention Estimation, ADAS (Autonomous Driver Assistance Systems), GazeCNN, Visual Transformers, DGAZE

1. Introduction

Attention while driving is a key element in road safety to keep passengers, drivers and pedestrians safe. Distractions caused by secondary tasks have been proved as the main factor in slowed responses in immediately dangerous situations [1], with 80% of reported crashes and 65% of near-crashes over 100 analyzed vehicles caused by unsafe driving behaviors such as inattention [2]. Moreover, the probability of collisions caused by driver distraction is significantly reduced in case passengers warn them about unseen hazards [3, 4]. This shows the importance of developing increasingly efficient Advanced Driver Assistance Systems (ADAS), especially with the use of artificial intelligence algorithms capable of understanding whether a driver is distracted from the road and alerting them. The identification of points of focus of drivers can also be used to train autonomous driving algorithms to pay more attention to some elements rather than to others, thus making them more capable of safe driving. Machine learning and distributed computing approaches e.g. cloud computing have become a cornerstone of modern data technology, playing a pivotal role in various sectors [5, 6]. In the green economy, machine learning algorithms help optimize energy consumption and reduce carbon footprint by predicting demand and managing supply efficiently. In the field of renewable energies, these algorithms aid in forecasting energy production from sources like wind and solar, thereby facilitating effective grid management. In the field of human-computer interaction, machine learning enhances user experience by enabling systems to understand and respond to human behavior in a more intuitive and personalized manner [7, 8, 9, 10, 11]. Lastly, in the automobile industry, machine learning is driving the revolution of autonomous vehicles and smart traffic management systems, contributing to safer and more efficient transportation [12]. The goal of this paper is to introduce a new approach to visual attention estimation for safe driving. To the best of our knowledge, most studies on driver attention are based either on the evaluation of driver behavior, without considering the environment surrounding the car, or exclusively on the road, training models to identify the elements to focus on. Our approach, in contrast, entails a comprehensive consideration of both the driver and the road views. Specifically, we assess the point of focus of the driver, contextually understanding whether they are paying attention to the road, and eventually which element of the road has captured their focus. To do this, we divide our task into two parts:

- Gaze point detection: we identify the point where the driver is looking at to assess where the driver is paying attention;
- Object identification: we identify the main ob-

SYSYEM 2023: 9th Scholar's Yearly Symposium of Technology, Engineering and Mathematics, Rome, December 3-6, 2023
fiani@diag.uniroma1.it (F. Fiani); samuele.russo@uniroma1.it (S. Russo); cnapoli@diag.uniroma1.it (C. Napoli)
0009-0005-0396-7019 (F. Fiani); 0000-0002-1846-9996 (S. Russo); 0000-0002-9421-8566 (C. Napoli)

(a) (b)

Figure 1: Example of paired images from the DGAZE dataset. (a) Driver view of driver number 22. (b) Sample 15 road view of driver number 22.

jects on the road, namely pedestrians, motorbikes, traffic signs, traffic lights, other cars, and trucks.

For the first task we will employ the GazeCNN model, a variant of a ResNet [13] that takes various facial features as input, such nose and left pupil position, head pose and eyes corners. In addition, to perform a comparative analysis between two different methods, we will also consider a Resnet+Transformer model [14] fine-tuned to output the exact position of where the driver is looking at. For the second task, instead, we use a fine-tuned YOLOv8 model, part of the YOLO family of object detection algorithms [15], configured to consider only the classes of interest. To accomplish our task, we used the DGAZE dataset [16], which to the best of our knowledge is one of the few dataset that provide both both the driver's view and the road view. This data was collected in a controlled laboratory setting where 112 street videos were projected in front of 20 'drivers', who were told to focus on a designated point annotated in the projected video. This dataset contains over 180,000 pairs of images, where each pair includes a road view and the corresponding driver view, plus a label indicating the coordinates of the point the driver was instructed to focus on (specifically, the center of the bounding box of the object). We reported an example of this dataset in Figure 1.

The paper is organized as follows. In Section 2, related works about gaze detection and driver gaze prediction are provided to frame our work in the current state-of-the-art scenario. Section 3 describes the data analysis, the pre-processing and feature extraction done on the DGAZE dataset and the proposed architecture to perform our task. Section 4 reports the performed experiments and the corresponding results. Finally, Section 5 presents the study's conclusions.

2. Related Works

2.1. Gaze Detection

Gaze detection is a highly significant topic in the field of Computer Vision and Human-Robot Interaction. Despite various advancements over time, it remains a challenging task due to aspects such as the uniqueness of faces and eyes, potential occlusions, differences in lighting, image quality, etc. Throughout literature, various methods have been employed, ranging from simple classification methods, like Random Forest [17] and SVM [18], to deep neural network models. The use of deep CNNs has greatly enhanced accuracy of this task, with great results obtained even with wild datasets [19]. While the majority of works use only the eyes to perform gaze estimation, other works use facial features different than the eyes, such as facial grids [20] or a combination of the eyes images and the head pose [21].

Transformers are also a viable novel solution, with two types of transformers derived from the Vision Transformer (ViT) framework finding success [14]. The first one, denoted as GazeTR-Pure, processes the cropped face as input, divides it into patches and passes them to a transformer encoder that will return the direction of gaze. In contrast, GazeTR-Hybrid adopts a hybrid approach, combining Convolutional Neural Networks (CNN) with transformers. The CNN extracts local feature maps from the face, which are then passed to the encoder transformer to capture the global relationships between the maps and finally obtain the desired output. These models take advantage of the transformer's attention mechanism to improve performances, with the GazeTR-Hybrid obtaining results comparable to the state-of-the-art. As previously mentioned GazeTR-Hybrid will be the base for one of our two approaches.

2.2. Driver Gaze Prediction

Driver gaze prediction task is approached in two ways in literature. The first approach focuses only on the interior images of the car (the driver's view) [22, 23, 24, 25]. Generally, the car is divided in different zones, such as the windscreen, the speedometer, the two side-view mirrors, the back mirror, and so on. The algorithms try to predict which of these areas the driver is looking at by analyzing the images of the driver.

The other approach, instead, is focused only on the outside the car. Many papers analyze images of the road recorded from inside the car via the windscreen to calculate an attention map, i.e. a heat map where brighter colors indicate the elements where drivers focus most while driving [26, 27, 28, 29]. Attention maps are extremely significant for autonomous driving, since they may be useful in training models that can understand, in a given driving situation, which of the many important elements of the road to focus on the most.

For what concerns the DGAZE dataset, already analyzed in the introduction, a related model called I-DGAZE has also been developed [16]. The model consists of two branches. The first is composed of a CNN with the ad-

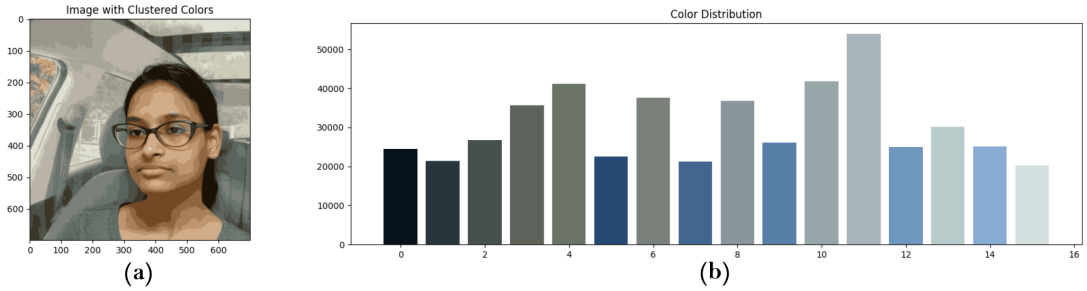

Figure 2: (a) Cropped driver 22 view subjected to K-Means clustering. (b) Corresponding color distribution histogram after K-clustering. On the x axis is represented the bin number, while on the y axis the number of pixel occurrences of the bin.

dition of a final flattened layer, which takes the driver's left eye as input. The other is composed of only dense layers and takes as input various features of the face, namely the pose, location, and area. The features generated by the two branches are then merged and passed to a fully connected layer that uses them to determine the coordinates of the gaze point (x, y).

Building on the literature just presented, our work is quite innovative in using an approach that is not widely used for the identification of drivers' attention while driving. It will also compare two models for gaze detection, as mentioned above, combining the results of these with those of YOLO in such a way as to output whether or not the driver is paying attention to the road and in particular to which element.

3. Materials and Methods

3.1. Data Analysis

Due to various challenges in gaze detection (e.g. eye-head interplay, illumination, eye registration errors, occlusions, difficulties in generalization of eye region appearance) [30], before proceeding with the implementation we conducted a thorough analysis of color distribution on our data, examining it in both RGB and HSV color spaces. Driver's view images were cropped to a 700 x 700 format from the top-left corner at pixelwise x and y coordinates (25, 100). This pre-processing step, consistently applied throughout our work, was designed with the specific purpose of eliminating non-essential areas within the image, focusing only on the face region.

We then computed histograms within the RGB and HSV color spaces for a randomly selected sample from each driver's image set. The K-Means algorithm was employed to cluster all the colors in 16 clusters, with the resulting histogram shown in Figure 2.

The relative 1D RGB graphs are presented in Figure 3, while the flattened 3D RGB graph is shown in Figure 4. Both graphs have been normalized to facilitate comparison. Finally, we selected three distance metrics to conduct a dataset-wide comparison between the histograms and computed the corresponding matrices:

- **Wasserstein (Earth Mover's) Distance:**

$$W(p,q) = \inf_{\gamma \in \prod(p,q)} \left(\int_{\mathbb{R}\times\mathbb{R}} ||x-y|| d\gamma(x,y) \right) \tag{1}$$

  where p and q are two probability distributions and $\prod(p,q)$ denotes the set of all joint probability distributions on $\mathbb{R} \times \mathbb{R}$ whose marginals are p and p. This metric is symmetric.
- **Chi-Squared Distance:**

$$\chi^2(p,q) = \sum_i \frac{(p(i)-q(i))^2}{p(i)} \tag{2}$$

  where p and q are two probability distributions.
- **Kullback-Leibler Divergence:**

$$D_{KL}(p||q) = \sum_{i\in\mathbb{R}} p(i)\, log\left(\frac{p(i)}{q(i)}\right) \tag{3}$$

  where p and q are two probability distributions on the same sample space $\mathbb{R}$.

The obtained matrices for 3D RGB graphs are reported in Figures 5, 6 and 7. Our data analysis indicate that there are no significant differences in color distribution among various driver images with the exception of certain drivers, such as Driver 13 and 5, with consistently high values among all metrics. Conversely, Drivers 2, 22, and 23 occasionally exhibit increased differences, but not consistently across all plots. The three metrics have also been calculated for 1D channels and averaged, producing similar results, therefore they will not be reported. The same process has also been repeated for the HSV color space, so the 1D and flattened 3D graphs have been computed, with an additional 2D heat map of the 3D

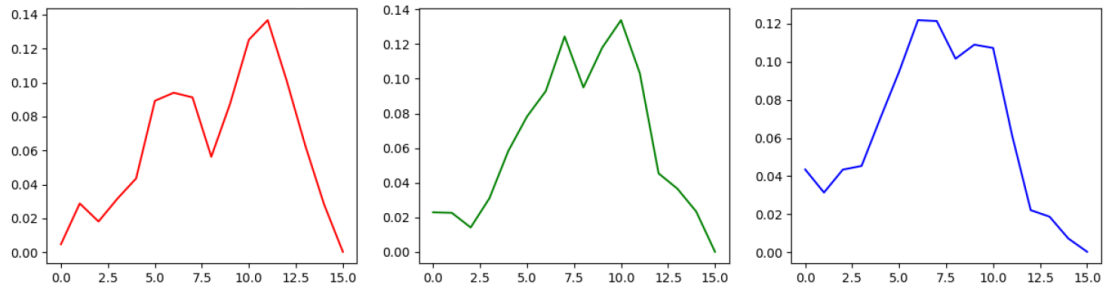

Figure 3: Graphs of the red, green and blue channel bin frequency distribution. Each channel has 16 bins (represented on the x axis), with the frequency for each bin represented on the y axis. The frequency distribution has been normalized.

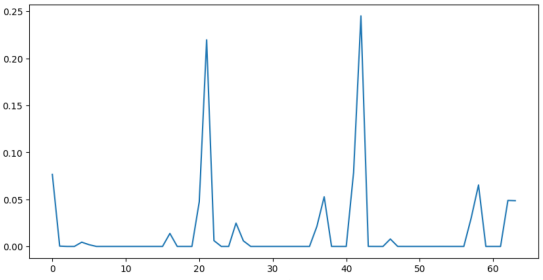

Figure 4: Graph of the flattened bin frequency distribution. 64 bins have been considered for the flattened 3D representation (represented on the x axis), with the frequency for each bin represented on the y axis. The frequency distribution has been normalized.

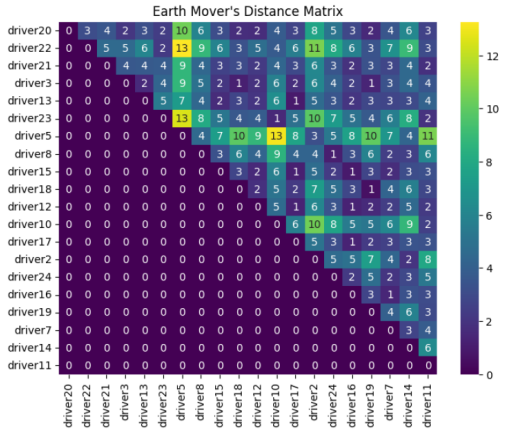

Figure 5: Wasserstein (Earth Mover's) distance matrix between all couples of sample drivers 3D color distribution. A high value indicates a big color space distance between images. Only the upper triangular matrix has been reported given the symmetry of the matrix.

graph, and the nine distance matrices have been computed. Given the use of RGB space during the experiments and the absence of significant differences in the HSV analysis, we will skip the presentation of the obtained results.

3.2. Architecture

As mentioned, our idea is to divide the model into two branches. The first branch predicts the exact coordinates (x, y) of the driver's focus point from the input driver view. The decision to predict the exact point of focus of the driver is due to the desire to achieve greater accuracy in estimating visual attention. This way, we will be able to distinguish precisely which element of the road they are paying more attention to even in case of elements overlapping. To the best of our knowledge, this is a situation that is not very usual in the literature and could be an important innovation to obtain increasingly accurate results in Visual Attention Estimation. The video lengths of view and driver videos were manually aligned since some view videos (which are common among all drivers) were mismatched in the number of frames with driver videos. The input is then processed to extract key components of the face, i.e. the driver's face, the left eye, the pupil position, the nose position, the head pose and the eye corners. A combination of SOTA tools for analyzing facial features was used: a shape predictor, obtained from dlib [31], for the extraction of the eyes and the position of the nose and pupils, a frontal face detector, also from dlib, for the extraction of the face, and SixDRepNet [32] for the extraction of the head pose.

Two types of models will be considered for this branch and confronted to evaluate the best one in terms of performance. The first model is GazeCNN, a variation in model and layers size of I-DGAZE. The model, shown in Figure 8, is composed of two branches which extract features used as inputs for the final fully connected layer. The first branch takes the cropped 3 x 32 x 64 left eye image

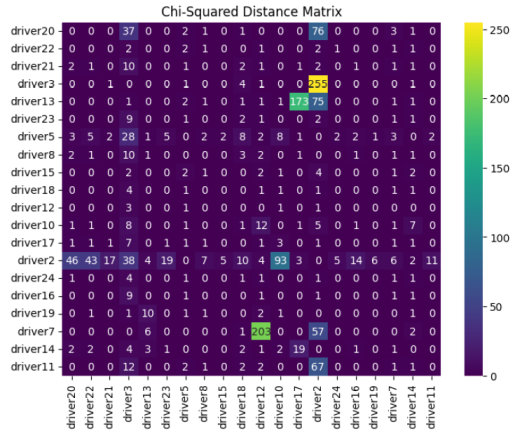

Figure 6: Chi-Squared distance matrix between all couples of sample drivers 3D color distribution. A high value indicates a big color space distance between images.

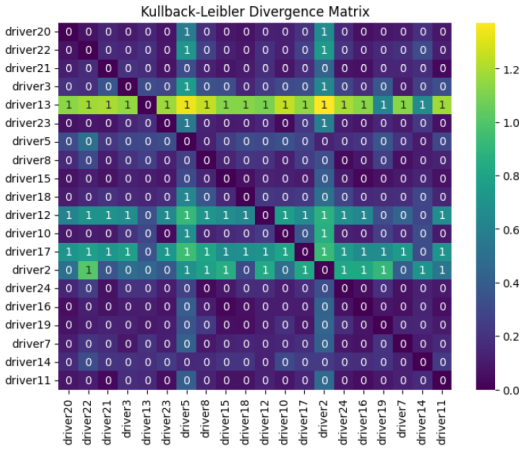

Figure 7: Kullback-Leibler divergence matrix between all couples sample drivers 3D color distribution. A high value indicates a big color space distance between images.

as input, which is then passed through three 8-channel convolutional layers. The second and third one are followed by a max-pooling layer each, while the second convolutional block has an additional residual connection compared to the original architecture. The resulting output is then flattened to obtain a 336-dimensional feature vector. The other branch, instead, takes a series of features as input. We examined two scenarios to assess the actual influence of features on the final outcome. In one case, we used a 7-dimensional face feature vector as input, comprising head pose and the positions of the two pupils, while in the other we also added the nose and eye corners positions. The performances for both scenarios

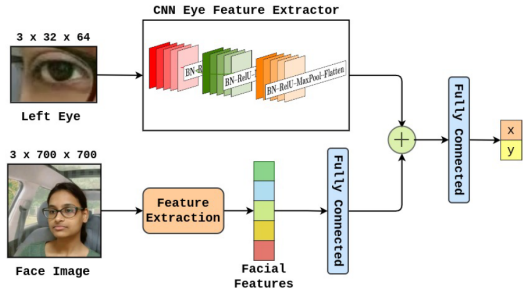

Figure 8: Schematic model of the GazeCNN architecture.

Table 1
Evaluation Metrics at best epoch in Test Dataset for the three selected models

Eye Feature Branch		
Layer	Kernel	Output Channels
Conv2D_1	3x3	8
Conv2D_2	3x3	8
MaxPool2d_1	4x4	8
Dropout		8
Conv2D_3	3x3	4
MaxPool2d_2	4x4	4
Flatten_1		336
Feature Branch		
Layer	Kernel	Output Channels
Dense_1		16
Fused Branch		
Layer	Kernel	Output Channels
Merge_1		352
Dense_2		64
Dense_3		2

will be discussed in the following section. This branch is composed of only a fully connected layer of output size 16. The two features vectors output from the branches are then merged in a 352-dimensional vector, which is then passed through two fully connected layers which output the final (x,y) coordinate vector of the driver's focus point. All the structure is summarized in Table 1.

The second model is GazeTR-Hybrid, composed of a ResNet which extracts local feature maps and a Visual Transformer which calculates global relationships between the feature maps and generates the gaze point. Our aim was to assess the performance of a transformer model in a domain where it is not commonly employed and to verify the applicability of GazeTR-Hybrid on a different task than the original (i.e. compute focus point instead of gaze direction). The original model with its pre-trained weights, but we performed fine-tuning to

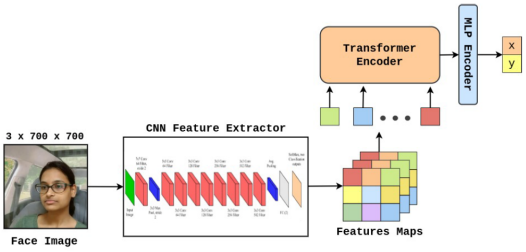

Figure 9: Schematic model of the GazeTR-Hybrid architecture.

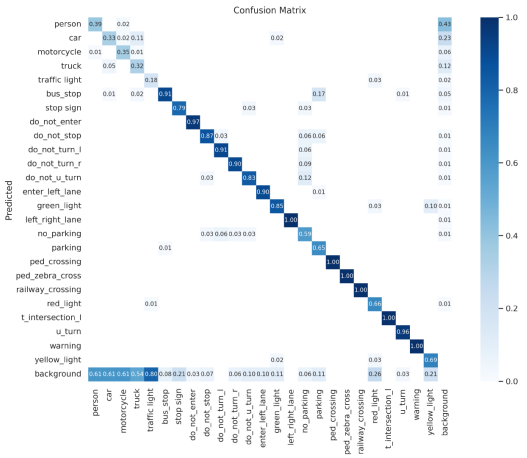

Figure 10: Confusion matrix of the YOLO fine-tuned model.

adapt the model for a direct confrontation with GazeCNN. The structure of GazeTR-Hybrid, shown in Figure 9, is composed of various convolutional layers, forming the ResNet-18 block, which generate $7 \times 7 \times 512$ feature maps from face images. The block is followed by an additional 1×1 convolutional layer aimed at adjusting the channel scale to obtain $7 \times 7 \times 32$ feature maps. The transformer block, instead, consists of six Transformer Encoder Layers which perform 8-heads self-attention mechanism, followed by a two-layer MLP with hidden size 512 and the dropout 0.1. The transformer is also equipped with a linear feedforward layer which produces the 2-dimensional output of the driver's gaze point.

The second branch performs object detection by passing as input to the model the various images of the 'road view' to recognize in each of them the most relevant elements. This is instrumental in identifying the most important objects on the road, those to which the driver should pay most attention to. For this purpose, we used a pre-trained YOLOv8 model, which was then fine-tuned on the elements that we were most interested in. This way, our fine-tuned YOLO model will be able to identify only the road elements of our interest while excluding irrelevant ones. For our task, we combined a dataset of road signs part of the RF100 initiative [33] with one created by us using images from the COCO dataset [34]. The images from COCO were carefully chosen to exclusively include pictures with the presence of people, cars, motorcycles, and trucks. This was done to prevent our fine-tuned YOLO model from forgetting these classes, which are crucial for our task. The other dataset, instead, contains various classes of road signs that were helpful for training YOLO to identify these road elements, which are the ones every driver should pay attention to. In total, we used 3589 images, divided into 2480 for the training set and 1109 for the validation set.

We fine-tuned the pre-trained YOLO model on this dataset for 40 epochs, resulting in a precision of 83.61%, a recall of 73.99%, and a mAP50 of 79.27%. We report in the Figure 10 the confusion matrix. We can see how our YOLO model performs quite well on almost all new classes of road signs, while its performance is lower in identifying cars, people, trucks and motorcycles. This could be attributed to the fact that in the images from the road signs dataset we only recognize one element of the considered class, leading to higher precision, whereas in the photos from COCO there are various elements of different classes in each image. This might lead to our fine-tuned model having more difficulty learning from images rich of different elements, resulting in poorer performance in those classes. We also see a particularly low precision for the traffic light class, probably influenced by the lower number of samples in our dataset. Despite this, for the use in our Visual Attention Estimation model, the achieved results can be considered acceptable.

The outputs of the two branches are finally combined to determine the final output of the model. If the driver's point of gaze falls within one of the bounding boxes of the road elements identified by YOLOv8, we can assert with confidence the driver's attention and identify which element they are looking at. In general, giving as input to our model a pair of images corresponding to the driver's view and the road view at a specific moment during driving (i.e. capturing what happens inside and outside the vehicle), it can determine whether the driver is paying attention to the road. Additionally, it can identify, and return in output, which specific element on the road is drawing more of the driver's interest at that moment. A schematic representation of the full defined model is shown in Figure 11.

4. Results and Discussion

To perform the experiments, the DGAZE dataset has been split into train set, validation set and test set according to the same original division [16]. Of the 20 drivers,

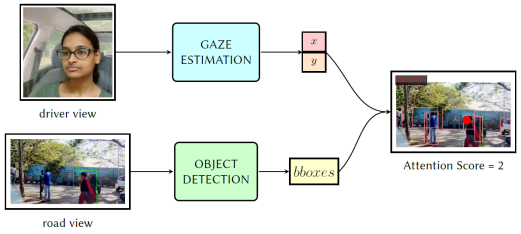

Figure 11: General Architecture presented in our paper. The network is divided in two branches: one which computes the point the driver focuses on, the other which identifies all the principal street objects. The model then assesses the driver's attention (whether they are looking at an element of the road) and which element they pay the most attention to.

16 were used for training (corresponding to 60% of the video sequences for training), 2 for validation (20%) and 2 were used for testing (20%). As mentioned earlier, various training experiments were conducted for both the GazeCNN and the GazeTR-Hybrid models. In addition, for the first model we also considered a scenario where the input also considers the position of the nose and the eye corners as features (i.e. a 17-feature vector) to assess whether increasing the number of features has any effect on the model's performance.

All the models were trained using L1 loss function, Adam optimizer with a learning rate of 1e-3, weight decay of 1e-5, β_1 = 0.9 and β_2 = 0.97. Additionally, a StepLR scheduler with a step size of 15000 and a gamma of 0.1 was also applied to improve training performance. The models were trained for 10 epochs with a batch size of 16. All the hyperparameters have been experimentally calculated to avoid overfitting and to reach the best performance possible. The experiments were performed using a NVIDIA GeForce RTX 3060 Laptop GPU. In the next subsection we will see more in details the results of these training experiments.

4.1. Driver Gaze Prediction

In this section, we will present the results of the experiments conducted for the gaze detection task. We consider the GazeCNN, the GazeCNN + features and the GazeTR-Hybrid (CNN + Transformer) models to perform this task. To validate results obtained, we consider three different metrics:

- **Accuracy w.r.t Threshold:**

$$acc_{tresh} = \frac{1}{n}\sum_{i\in\mathcal{I}} x_i \tag{4}$$

where $\mathcal{I}$ is the set of images in the dataset, $n = |\mathcal{I}|$ the cardinality of the set and

$$x_i = \begin{cases} 1 & \text{if } d(g_i, \hat{g}_i) < \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $d(g_i, \hat{g}_i) = \sqrt{(g_i - \hat{g}_i)^2}$ is the Euclidean distance, $\hat{g}_i$ is the estimated gaze point and g_i the true gaze point in the road view image coordinates. The threshold has been set to 250 pixels.

- **Accuracy w.r.t Bounding Box:**

$$acc_{bbox} = \frac{1}{n}\sum_{i\in\mathcal{I}} x_i \tag{6}$$

where $\mathcal{I}$ is the set of images in the dataset, $n = |\mathcal{I}|$ the cardinality of the set and

$$x_i = \begin{cases} 1 & \text{if } g_i \in \text{boundingbox} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $\hat{g}_i$ is the estimated gaze point and g_i the true gaze point in the road view image coordinates. The bounding box considered is the one surrounding the road element that, during the creation of the dataset, is observed by drivers.

- **Displacement via Euclidean Distance:**

$$D(g_i, \hat{g}_i) = \frac{1}{n}\sum_{i=1}^{n} \sqrt{(g_i - \hat{g}_i)^2} \tag{8}$$

where $\hat{g}_i$ is the estimated gaze point and g_i the true gaze point in the road view image coordinates.

Table 2 shows the evaluation of the three metrics in the three selected models at the best epoch during the testing phase. The CNN + Transformer model performs better compared to the GazeCNN model in all cases. This demonstrates the effectiveness of this model in the considered task. We believe that, with an increase in epochs and input features, the CNN + Transformer model has the potential to achieve even better results by increasing the accuracy in calculating the driver's point of gaze. Instead, regarding the GazeCNN + features and the CNN + Transformer, we can observe that the latter proves to be superior in both bounding box accuracy and Euclidean error, while the former slightly outperforms in threshold accuracy. We can observe how the addition of input features (eye corners and nose position) leads to a remarkable improvement in performance for GazeCNN, proving to be a crucial factor in the learning process.

We would like to point out that, for all the models, the bounding box (bbox) accuracy is relatively low. This can be explained by the fact that, for many videos in the dataset, the fixation elements tend to be small, as they

Table 2
Evaluation Metrics at best epoch in Test Dataset for the three selected models

Model	Threshold Accuracy [%]	Bbox Accuracy [%]	Euclidean Error [px]
GazeCNN	37.57	15.97	371.93
GazeCNN + features	46.33	18.50	320.54
CNN + Transformer	45.62	19.72	317.40

Table 3
Comparison table between our models and other SOTA eye gaze models on train, validation and test pixel accuracy (calculated via Mean Absolute Error)

Model	Train Error [px]	Val Error [px]	Test Error [px]
Turker Gaze [35]	171.30	176.37	190.72
iTracker [20]	140.10	205.65	190.5
I-DGAZE [16]	133.34	204.77	186.89
GazeCNN	163.00	154.41	228.46
GazeCNN + features	171.99	174.63	199.99
CNN + Transformer	200.85	197.88	196.53

are far away, and therefore the corresponding bounding boxes are similarly small. Accuracy for bounding boxes is very restrictive, since the presence of an error, even by a single pixel, could cause the point to be outside the corresponding bounding box and therefore lead to a decrease in the accuracy.

Considering the analyzed results, the GazeTR-Hybrid (CNN + Transformer) model has been employed in the overall Driver Visual Attention Estimation model to perform point-gaze estimation. To confirm what has been discussed so far, we present a comparison in Table 3 between the models just considered and some SOTA eye gaze models. In particular, we consider the model proposed in TurkerGaze [35], where they use pixel-level face features as input and use Ridge Regression to estimate gaze point on the screen, the one proposed in Eye-tracking for Everyone [20], which predicts user gaze on phone and tablet, and finally I-DGAZE, the model presented in our reference paper [16].

The error used as a metric for this comparison is the Mean Absolute Error (MAE), calculated by taking the mean of the absolute differences between model predictions and actual values. In mathematical terms, it is expressed as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|g_i - \hat{g_i}| \tag{9}$$

where n is the total number of samples, g_i represents the actual values and $\hat{g_i}$ represents the model predictions. The smaller the Mean Absolute Error, the more accurate the model is in predicting the co-ordinates of the gaze point. We can see that even in this case the CNN + Transformer model proves to be in line with the other SOTA models on the validation and test error, proving the efficacy of the method. In contrast, the train error is the highest. This phenomenon does not fit with any classical training schema and is therefore not correlated to underfitting or overfitting, but a lower validation error compared to train error may be caused by the samples selected for validation being particularly simple to predict for the network. Finally, it is important to note that the GazeCNN model has the lowest validation error. However, this is associated with a higher test error, possibly indicating overfitting during training.

4.2. Driver Attention Evaluation

In Table 4 we describe the results obtained from the analysis of drivers' attention using the general model described by the Figure 11. To perform this analysis, we considered only the two drivers belonging to the test set as specified above out of the total 20 included in the dataset. The dataset used, DGAZE, provides bounding boxes coordinates as labels only corresponding to the object observed by the driver. Therefore, we have considered these bounding boxes as indicative of the most important element in the scene, and we will consider any detected object aside from the selected one as an incorrect focus object. Based on this reasoning, we identified three attention score scenarios:

- Correct bbox (Attention Score=2): the driver is looking at the correct road element indicated by the dataset, so the point the driver is focusing on falls in the bounding box of the expected object;
- Another bbox (Attention Score=1): the driver is attentive, but focused on an another element of the road, so the point the driver is focusing on

Table 4
Results of Visual Attention Estimation in Drivers. An attention score of 2 indicates a correct object of focus, an attention score of 1 an incorrect object of focus but an attentive driver and an attention score of 0 a distracted driver.

Attention Score	Percentage [%]
Correct bbox (Att. Score = 2)	16.06
Another bbox (Att. Score = 1)	29.95
No bbox (Att. Score = 0)	53.99

falls in the bounding box of an object different from the one of the expected object;

- No bbox (Attention Score=0): the driver is not paying attention to the road and is therefore not looking at any important road elements, so the point the driver is focusing on doesn't fall in any bounding box.

We observe that the system identifies distracted drivers (Attention Score=0) 53.99% of the time, a percentage which does not fall in line with expected results. Unfortunately, this result is attributed to the suboptimal performance of our CNN + Transformer model, particularly in bbox accuracy which as shown in Table 4 is particularly low (less than 20%). As mentioned earlier, this is a challenging task, as even small pixel errors in this context have significant relevance, and it therefore highlights the need for greater precision in determining the gaze point, especially in such cases where a high accuracy is necessary due to safety reasons.

In the scenario where the system recognizes drivers as attentive, instead (approximately 46.01% of the time), we notice that generally they are attentive but focused on road elements that are not considered the most important (Attention Score=1). The data presented in Table 5 reveals that, most often, drivers concentrate their attention on the vehicles in front of them, especially on cars and trucks. This indicates a higher level of attention to other vehicles compared to road signs or other objects, which justifiable due to other vehicles being the main 'antagonistic' driving element and the primary source of potential impediment to road safety. Even though in our dataset we have predetermined attention objectives, which consequently limits the correctness of the obtained results, a statistical analysis can be performed with our framework in different scenarios to gain insight on drivers' attention behaviour and on the objects that they pay most attention to in different driving situations.

Table 5
Object focus distribution in test set for drivers. Obtained data shows that drivers tend to focus their attention on vehicles (car and truck) compared to other elements.

Object Type	Percentage [%]
person	8.33
truck	15.70
car	16.40
road signal	2.90
motorcycle	2.66
traffic light	0.01

5. Conclusion

In this paper we presented a new way to perform the task of driver visual attention detection. As already mentioned, this is obtained by performing two sub-tasks, gaze estimation and the object detection. To execute the first, we examined two different architectures, GazeCNN and GazeTR-Hybrid. We then assessed the performance of both models for the specified task, achieving better results with the GazeTR-Hybrid model. This second model was consequently used to implement driver visual attention detection. For object detection, we employed a fine-tuned YOLOv8 model capable of recognizing cars, people, trucks, motorcycles, traffic lights and various road signs. By combining the outputs of the two branches, i.e. projecting the driver's gaze point (whose coordinates are obtained as output from the gaze detection branch) onto the corresponding 'road view', where all relevant road objects identified by YOLO are located, we evaluated the actual visual attention of drivers. This approach allowed us to obtain two valuable pieces of information: whether the driver is attentive or not and, if so, to which element of the road.

Possible future improvements are evident, starting with the gaze detection task, where increased precision in calculating the gaze point could lead to better results in assessing drivers' visual attention. We believe that the addition of more features during the training phase to the GazeTR-Hybrid model could lead to the desired improvement in performances, thus achieving increasingly precise results. This, in turn, would contribute to an effective improvement in Visual Attention Estimation in drivers. This is a consequence of the fact that, by increasing precision, we can identify information about the objects the driver is focusing on even in case of occlusions, i.e. if they are distant or partially hidden by other elements. However, we find our approach to the Driver Vision Attention task promising for future works, particularly in the aspect of obtaining more complete results on the drivers' engagement with the road.

Drivers' attention and the object they focus on can be subsequently used in different contexts. For instance, the former could be applied in assessing attention in systems designed to alert the driver when not paying adequate attention to the road, while the second can be

used to train autonomous driving models, helping them understand what to prioritize in each driving scenario. A mixed model able to detect both data could lead to more comprehensive autonomous or assisted driving systems by reducing training times due to faster data collection.

References

[1] A. Eriksson, N. A. Stanton, Takeover time in highly automated vehicles: noncritical transitions to and from manual control, Human factors 59 (2017) 689–705.

[2] T. A. Dingus, S. G. Klauer, V. L. Neale, A. Petersen, S. E. Lee, J. Sudweeks, M. A. Perez, J. Hankey, D. Ramsey, S. Gupta, C. Bucher, Z. R. Doerzaph, J. Jermeland, R. R. Knipling, The 100 car naturalistic driving study: Phase II – Results of the 100-car field experiment (2006).

[3] T. Rueda-Domingo, P. Lardelli-Claret, J. de Dios Luna-del Castillo, J. J. Jiménez-Moleón, M. Garciá-Martiń, A. Bueno-Cavanillas, The influence of passengers on the risk of the driver causing a car collision in spain: Analysis of collisions from 1990 to 1999, Accident Analysis & Prevention 36 (2004) 481–489.

[4] K. A. Braitman, N. K. Chaudhary, A. T. McCartt, Effect of passenger presence on older drivers' risk of fatal crash involvement, Traffic injury prevention 15 (2014) 451–456.

[5] F. Bonanno, G. Capizzi, G. L. Sciuto, C. Napoli, G. Pappalardo, E. Tramontana, A novel cloud-distributed toolbox for optimal energy dispatch management from renewables in igss by using wrnn predictors and gpu parallel solutions, 2014, pp. 1077 – 1084. doi:10.1109/SPEEDAM.2014.6872127.

[6] I. E. Tibermacine, A. Tibermacine, W. Guettala, C. Napoli, S. Russo, Enhancing sentiment analysis on seed-iv dataset with vision transformers: A comparative study, 2023, pp. 238 – 246. doi:10.1145/3638985.3639024.

[7] N. N. Dat, V. Ponzi, S. Russo, F. Vincelli, Supporting impaired people with a following robotic assistant by means of end-to-end visual target navigation and reinforcement learning approaches, volume 3118, 2021, pp. 51 – 63.

[8] V. Ponzi, S. Russo, A. Wajda, R. Brociek, C. Napoli, Analysis pre and post covid-19 pandemic rorschach test data of using em algorithms and gmm models, volume 3360, 2022, pp. 55 – 63.

[9] A. Alfarano, G. De Magistris, L. Mongelli, S. Russo, J. Starczewski, C. Napoli, A novel convmixer transformer based architecture for violent behavior detection 14126 LNAI (2023) 3 – 16. doi:10.1007/978-3-031-42508-0_1.

[10] E. Iacobelli, V. Ponzi, S. Russo, C. Napoli, Eye-tracking system with low-end hardware: Development and evaluation, Information (Switzerland) 14 (2023). doi:10.3390/info14120644.

[11] F. Fiani, S. Russo, C. Napoli, An advanced solution based on machine learning for remote emdr therapy, Technologies 11 (2023). doi:10.3390/technologies11060172.

[12] N. Brandizzi, S. Russo, G. Galati, C. Napoli, Addressing vehicle sharing through behavioral analysis: A solution to user clustering using recency-frequency-monetary and vehicle relocation based on neighborhood splits, Information (Switzerland) 13 (2022). doi:10.3390/info13110511.

[13] 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.

[14] Y. Cheng, F. Lu, Gaze estimation using transformer, in: 2022 26th International Conference on Pattern Recognition (ICPR), IEEE, 2022, pp. 3341–3347.

[15] J. Terven, D.-M. Córdova-Esparza, J.-A. Romero-González, A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas, Machine Learning and Knowledge Extraction 5 (2023) 1680–1716.

[16] I. Dua, T. A. John, R. Gupta, C. Jawahar, Dgaze: Driver gaze mapping on road, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 5946–5953.

[17] 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.

[18] D. Melesse, M. Khalil, E. Kagabo, T. Ning, K. Huang, Appearance-based gaze tracking through supervised machine learning, in: 2020 15th IEEE International Conference on Signal Processing (ICSP), volume 1, IEEE, 2020, pp. 467–471.

[19] X. Zhang, Y. Sugano, M. Fritz, A. Bulling, Appearance-based gaze estimation in the wild, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4511–4520.

[20] 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.

[21] T. Fischer, H. J. Chang, Y. Demiris, Rt-gene: Real-time eye gaze estimation in natural environments, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 334–352.

[22] H. S. Yoon, N. R. Baek, N. Q. Truong, K. R. Park, Driver gaze detection based on deep residual networks using the combined single image of dual

near-infrared cameras, IEEE Access 7 (2019) 93448–93461.

[23] N. Mizuno, A. Yoshizawa, A. Hayashi, T. Ishikawa, Detecting driver's visual attention area by using vehicle-mounted device, in: 2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC), IEEE, 2017, pp. 346–352.

[24] S. Vora, A. Rangesh, M. M. Trivedi, Driver gaze zone estimation using convolutional neural networks: A general framework and ablative analysis, IEEE Transactions on Intelligent Vehicles 3 (2018) 254–265.

[25] S. M. Shah, Z. Sun, K. Zaman, A. Hussain, M. Shoaib, L. Pei, A driver gaze estimation method based on deep learning, Sensors 22 (2022) 3959.

[26] T. Deng, H. Yan, L. Qin, T. Ngo, B. Manjunath, How do drivers allocate their potential attention? driving fixation prediction via convolutional neural networks, IEEE Transactions on Intelligent Transportation Systems 21 (2019) 2146–2154.

[27] Y. Xia, D. Zhang, A. Pozdnoukhov, K. Nakayama, K. Zipser, D. Whitney, Training a network to attend like human drivers saves it from common but misleading loss functions, arXiv preprint arXiv:1711.06406 (2017).

[28] C. Gou, Y. Zhou, D. Li, Driver attention prediction based on convolution and transformers, The Journal of Supercomputing 78 (2022) 8268–8284.

[29] A. Palazzi, D. Abati, F. Solera, R. Cucchiara, et al., Predicting the driver's focus of attention: the dr (eye) ve project, IEEE transactions on pattern analysis and machine intelligence 41 (2018) 1720–1733.

[30] S. Ghosh, A. Dhall, M. Hayat, J. Knibbe, Q. Ji, Automatic gaze analysis: A survey of deep learning based approaches, IEEE Transactions on Pattern Analysis and Machine Intelligence 46 (2023) 61–84.

[31] D. E. King, Dlib-ml: A machine learning toolkit, The Journal of Machine Learning Research 10 (2009) 1755–1758.

[32] T. Hempel, A. A. Abdelrahman, A. Al-Hamadi, 6d rotation representation for unconstrained head pose estimation, in: 2022 IEEE International Conference on Image Processing (ICIP), IEEE, 2022, pp. 2496–2500.

[33] R. 100, road signs dataset, https://universe.roboflow.com/roboflow-100/road-signs-6ih4y, 2023. URL: https://universe.roboflow.com/roboflow-100/road-signs-6ih4y.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.

[35] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.