Bi-Model Architecture for Gaze Estimation in Visual Attention Analysis

1st Stanislava Kozakijevic
Faculty of Philosophy
University of Novi Sad
Novi Sad, Serbia
0009-0004-4927-4632

2nd Luka Jovanovic
Faculty of Technical Sciences
University of Pristina in Kosovska Mitrovica
Kosovska Mitrovica, Serbia
0000-0001-9402-7391

3rd Petar Spalevic
Faculty of Technical Sciences
University of Pristina in Kosovska Mitrovica
Kosovska Mitrovica, Serbia
0000-0002-6867-7259

4th Gradimirka Popovic
Department Pec-Leposavic
Kosovo and Metohija Academy of Applied Studies
Leposavic, Serbia
gradimirka.popovic@akademijakm.edu.rs

5th Bata Vasic
Faculty of Business and Law
MB University
Belgrade, Serbia
bata.vasic@ppf.edu.rs

6th Bogdan Ignjatovic
Exeshop DOO
Nis, Serbia
0009-0004-4425-1378

***Abstract*—The human gaze reveals attention, interests, and cognitive processing, yet real-time tracking on consumer-grade devices remains challenging. We present a bi-model approach for gaze estimation using a single webcam, optimized for efficient browser-based execution. A paired model training methodology is introduced, including a moving-target calibration procedure that maps iris positions to screen coordinates. Multiple architectures are evaluated, and recurrent models, particularly Gated recurrent networks (GRUs), are selected for their balance of accuracy, sequence modeling capability, and low computational demands. Experimental results show that the system can reliably estimate gaze in real time while maintaining accessibility and low hardware requirements. This framework provides a scalable, low-cost solution for eye-tracking research and interactive applications, enabling broad deployment without specialized equipment.**

***Index Terms*—Eye tracking, Gaze tracking, Recurrent neural networks, Machine learning, Computer vision.**

I. Introduction

The direction a person is looking in yields a plethora of information and has therefore been an established area of research in fields that inspect human behavior and cognitive processes. The field of language psychology has produced an Eye-mind hypothesis [1], based on empirical evidence that the reader's gaze lingers on parts of text with particularly important or cognitively demanding information. Another area of research where eye gaze tracking is crucial is developmental psychology, when working with infants, allowing for better understanding of cognitive functioning in preverbal subjects [2]. Gaze tracking is a method that allows for more objective measure of, sometimes unconscious, cognitive processes in participants, and is thus highly valued.

However, precise eye tracking can be difficult due to the nature of eye movement, including saccades which are rapid and frequent gaze jumps, and fixations which are brief periods of stability of gaze, as well as due to head movements, eye shape individual differences as well as glasses.

A variety of eye tracking solutions are currently available, ranging from high-precision infrared systems to wearable devices and software-based webcam approaches. Laboratory-grade systems provide accurate measurements of gaze and facial features, but their cost, specialized hardware requirements, and controlled experimental setup can restrict sample sizes and ecological validity. Wearable devices allow for more naturalistic tracking but still require dedicated equipment and calibration procedures.

To address these limitations, a bi-model gaze estimation framework was designed, implemented via browser and any standard web camera. A short training phase that each participant goes through overcomes the issue of individual differences. After evaluating multiple neural architectures, gated recurrent networks were selected due to their favorable trade-off between accuracy, temporal modeling capacity, and computational efficiency. By eliminating the need for specialized hardware, this approach offers a low-cost and scalable alternative for eye-tracking research and interactive applications.

The contributions of this work can be summarized as follows:

- A bi-model architecture for human gaze estimation.
- A lightweight training methodology designed for accessible public use.
- A web-based application for tracking participant gaze using recurrent neural networks and publicly available face mesh models.

II. Related works

Most used eye tracking technologies currently tend to require specialized hardware or software setup, such as Infrared video-based systems (e.g., EyeLink, Tobii Pro), and wearable systems (e.g., Tobii Pro Glasses, Pupil Labs). These options provide high spatial and temporal resolution, but tend to require laboratory settings, limiting the sample size and generalization [3].

Artificial neural networks (ANNs) have become highly prominent in industry due to their ability to model complex, non-linear relationships and learn patterns from large datasets. They have been successfully applied across diverse domains, including healthcare [4], IoT security [5], and phishing email detection [6].

Sequential models are neural networks specifically designed to process data that comes in sequences, where the order of input is crucial for correct analysis of the data. These models have been successfully implemented in flood prediction [7], medicine [8], [9], energy consumption prediction [10], [11].

This work addresses existing gaps in gaze estimation literature by introducing a bi-model architecture for human gaze prediction, coupled with a lightweight training methodology designed for accessible public use. In addition, it provides a web-based application capable of tracking participant gaze in real time using recurrent neural network architectures and publicly available face mesh models. Together, these contributions offer a practical, low-cost, and computationally efficient solution for expanding gaze data collection in public-facing studies.

A. Models utilized in this study

Artificial neural networks (ANNs) [12] are computational models inspired by biological neural systems, consisting of interconnected units that adjust their connection weights during training. They can learn complex, non-linear relationships between inputs and outputs, making them suitable for tasks such as image recognition, time-series prediction, and gaze estimation.

Long-short term memory neural networks (LSTM) [13] is an enhanced form of recurrent neural network (RNN) designed to address common issues such as vanishing and exploding gradients. LSTMs feature specialized memory cells that retain information across time steps, allowing the network to capture long-term dependencies in sequential data. Each memory cell contains three gates: forget, input, and output gates, that regulate the flow of information into, through, and out of the cell, enabling selective updating and retention of past states. Through this gating mechanism, LSTMs can maintain relevant information over extended sequences while discarding unnecessary or outdated data.

Gated Recurrent Units (GRUs) [14] are a streamlined variant of recurrent neural networks designed to process sequential data efficiently and mitigate the vanishing gradient problem. GRUs use two gates: the update and reset gates to control how much past information is retained or forgotten at each time step. This simpler architecture makes GRUs faster to train and computationally lighter than LSTMs while still capturing long-term dependencies. They have been successfully applied in time-series prediction, language processing, speech recognition, and, in this work, modeling temporal dynamics in gaze estimation.

Transformers [15] are neural networks designed for sequential data that use self-attention to capture relationships across a sequence without recurrence. This allows them to model long-range dependencies efficiently while enabling parallelized training. They have achieved exceptional results in language processing, time-series prediction, and image analysis, gaining popularity and earning them a place in the current research.

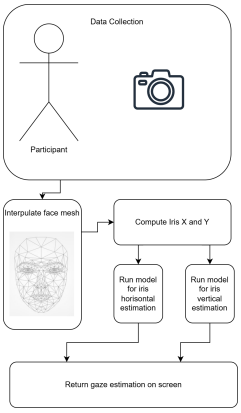

Fig. 1: Bi-model gaze estimation flowchart.

III. Methodology

The gaze estimation process is summarized in Algorithm 1. Each frame captured from the webcam is first processed using the MediaPipe Face Mesh [16] framework to detect facial landmarks, including precise iris positions. These iris coordinates are then normalized relative to each eye and input into a pair of estimation models—one predicting the horizontal (X) coordinate and the other predicting the vertical (Y) coordinate of the gaze. The resulting screen coordinates are logged continuously for analysis and visualization, enabling real-time, continuous gaze tracking.

Algorithm 1 : Interpolation Algorithm

```
Capture image from webcam
Apply face mesh model
Extract iris data
Interpolate X and Y position of each iris (relative to eye)
Apply estimation model for X
Apply estimation model for Y
Log position
```

Figure 1 shows a high-level flowchart of the bi-model gaze estimation procedure, illustrating the parallel processing of horizontal and vertical gaze predictions.

A. Training Procedure

Training of the estimation models is performed using a controlled visual stimulus procedure, designed to generate gaze data. Algorithm 2 summarizes the steps of this procedure. Participants are shown a static image while a yellow circle (radius $\approx$ 10 px) is displayed at random positions. The circle is held in place until the participant's gaze stabilizes for at least 2 seconds, after which it moves according to a randomly selected motion profile (quick–slow, slow–quick–slow, or slow–quick). During motion, iris positions are captured, normalized, and used alongside ground truth target coordinates to compute

Fig. 2: Visualization of the gaze training procedure. Left: comparison of estimated and ground truth gaze positions. Right: arrows illustrate gaze correction updates.

prediction errors. Model weights are updated via gradient descent to minimize these errors.

Algorithm 2 : Training Procedure for Gaze Estimation Models

```
Initialize estimation models for X and Y
Initialize model weights and biases
for each training iteration do
    Display a static image to the user
    Render a yellow circle (radius ≈ 10 px) at a random screen position
    Hold circle position until gaze stabilizes for at least 2 seconds
    Select a motion profile at random:
        (quick–slow), (slow–quick–slow), or (slow–quick)
    Move the circle to a new random position using the selected acceleration curve
    Capture facial image from webcam
    Apply face mesh model
    Extract iris landmarks
    Compute normalized iris X and Y positions (relative to eye)
    Predict screen coordinates using estimation models
    Compute loss between predicted position and ground truth circle position
    Update weights and biases of X and Y models using gradient descent
end for
```

Figure 2 provides a visual depiction of the training process. The left panel shows the estimated gaze location alongside the ground truth target, while the right panel illustrates the gaze correction process that iteratively refines model predictions.

In practice, the propose method is implemented as a web-based application, allowing real-time inference without specialized hardware. Lightweight neural network architectures are employed to balance prediction accuracy with low computational demand. The system is capable of continuously logging gaze positions while participants interact with visual stimuli, enabling scalable data collection for public-facing studies.

IV. Experimental Setup

To facilitate experimentation a brief simulation is conducted. A total of approximate 5 min of eye moments are captured using a python script developed specifically for this study. Ground truth samples are generated randomly using a yellow marker moved across the screen. Webcam images of the participants face are recorded and processed using a publicly available MediaPipe Face Mesh [16] model to extract face meshes and iris location. Based on iris location, face position and ground truth values, interpellation models are trained and evaluated using standard regression metrics including [17] with formulas provided in Eq. 1 to Eq 4.

Model	MAE	MSE	RMSE	R^2
LinearRegression	0.557092	0.478883	0.692013	0.515725
XGBoost	0.404003	0.287094	0.535811	0.708743
ANN	0.098294	**0.016946**	**0.130177**	0.709982
GRU	0.098419	0.017301	0.131534	0.705182
LSTM	**0.096954**	0.017395	0.131890	**0.704701**
Transformer	0.164302	0.041589	0.203935	0.286479

TABLE I: Model comparison in terms of standard regression metrics.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|p_i - \hat{p}_i| \tag{1}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(p_i - \hat{p}_i)^2 \tag{2}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - \hat{p}_i)^2} \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(p_i - \hat{p}_i)^2}{\sum_{i=1}^{n}(p_i - \bar{p})^2} \tag{4}$$

in the above equations, p_i is the true value, $\hat{p}_i$ is the predicted value, $\bar{p}$ is the mean of the true values, and n is the total number of samples.

A set of interpolation approaches was evaluated under identical conditions to assess their ability to map excitations to ground truth values. The tested algorithms include linear regression, extreme gradient boosting [18], and several neural network architectures such as vanilla artificial neural networks (ANN) [12], gated recurrent units (GRU) [14], long short-term memory networks (LSTM) [13], and a transformer-based architecture [15]. All algorithms were implemented using consistent default hyperparameter settings, and the neural networks were configured with two hidden layers, each containing 25 neurons.

V. Simulation Outcomes

Table I presents a comparison of multiple models for gaze position prediction using standard regression metrics, including MAE, MSE, RMSE, and R². Linear regression and XGBoost show moderate performance, whereas neural network models (ANN, GRU, LSTM) achieve substantially lower errors, with LSTM slightly outperforming GRU in MAE and RMSE. Although GRU does not achieve the absolute best numerical scores, it offers a favorable balance of accuracy, computational efficiency, and the ability to model sequential dependencies, making it the preferred choice for the study. The Transformer model underperforms relative to the recurrent networks, highlighting the suitability of sequence-based architectures for this type of gaze-tracking task.

Figure 3 shows the density of gaze points across the screen during the calibration task. The left panel displays the ground truth positions collected from the user's eye movements, while the right panel shows the GRU model's predicted gaze positions. Comparing the two heatmaps highlights areas where

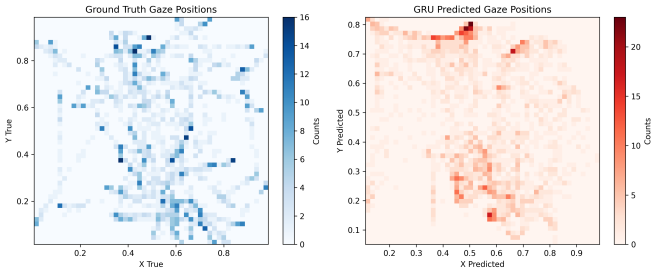

Fig. 3: Comparison of ground truth gaze positions (left) and GRU-predicted gaze positions (right).

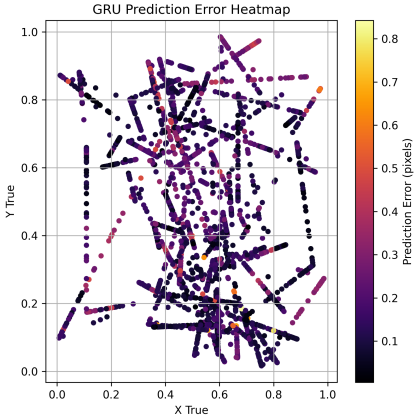

Fig. 4: Prediction error heatmap.

the model accurately captures user attention and where it deviates.

This heatmap shown in Figure 4 depicts the spatial distribution of prediction errors across the screen. Higher color intensity indicates regions where the GRU model has larger deviations from the ground truth. It allows identification of areas where the model performs poorly and highlights patterns in the prediction errors.

VI. Conclusion

This work presents an ultra low-cost and computationally efficient methodology for on-screen human gaze estimation, leveraging openly available tools. A bi-model architecture is proposed, along with a training methodology designed to build a more robust interpolation approach. The methodology is implemented in a web application intended for public-facing studies, enabling more accurate data collection and higher participant counts through increased convenience. The study explores several sequential models, with a focus on GRU networks for interpolating gaze positions on images. Promising results are observed using simple, default models; however, only rudimentary architectures with default hyperparameters are considered in this study.

Future work aims to expand upon this foundation, applying hyperparameter optimization techniques based on metaheuristics to improve model performance while minimizing computational demand. Additionally, development will target mobile devices to further increase accessibility and participant reach.

References

[1] M. A. Just and P. A. Carpenter, "A theory of reading: from eye fixations to comprehension.," *Psychological review*, vol. 87, no. 4, p. 329, 1980.

[2] R. L. Fantz and S. Nevis, "Pattern preferences and perceptual-cognitive development in early infancy," *Merrill-Palmer Quarterly of Behavior and Development*, vol. 13, no. 1, pp. 77–108, 1967.

[3] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer, *Eye tracking: A comprehensive guide to methods and measures*. oup Oxford, 2011.

[4] M. Gajevic, N. Milutinovic, J. Krstovic, L. Jovanovic, M. Marjanovic, and C. Stoean, "Artificial neural network tuning by improved sine cosine algorithm for healthcare 4.0," in *Proceedings of the 1st international conference on innovation in information technology and business (ICI-ITB 2022)*, vol. 104, p. 289, Springer Nature, 2023.

[5] L. Jovanovic, "Performance of sine cosine algorithm for ann tuning and training for iot security," in *Hybrid Intelligent Systems: 22nd International Conference on Hybrid Intelligent Systems (HIS 2022), December 13–15, 2022*, vol. 647, p. 302, Springer Nature, 2023.

[6] B. Lakicevic, Z. Spalevic, I. Volas, L. Jovanovic, M. Zivkovic, T. Zivkovic, and N. Bacanin, "Artificial neural networks with soft attention: Natural language processing for phishing email detection optimized with modified metaheuristics," in *International Conference on Advanced Network Technologies and Intelligent Computing*, pp. 421–438, Springer, 2024.

[7] I. Markovic, J. Krzanovic, L. Jovanovic, A. Toskovic, N. Bacanin, A. Petrovic, and M. Zivkovic, "Flood prediction based on recurrent neural network time series classification boosted by modified metaheuristic optimization," in *International Conference on Advances in Data-driven Computing and Intelligent Systems*, pp. 289–303, Springer, 2023.

[8] F. Markovic, L. Jovanovic, P. Spalevic, J. Kaljevic, M. Zivkovic, V. Simic, H. Shaker, and N. Bacanin, "Parkinsons detection from gait time series classification using modified metaheuristic optimized long short term memory," *Neural Processing Letters*, vol. 57, no. 1, p. 14, 2025.

[9] B. Radomirovic, N. Bacanin, L. Jovanovic, V. Simic, A. Njegus, D. Pamucar, M. Köppen, and M. Zivkovic, "Optimizing long-short term memory neural networks for electroencephalogram anomaly detection using variable neighborhood search with dynamic strategy change," *Complex & Intelligent Systems*, vol. 10, no. 6, pp. 7987–8009, 2024.

[10] L. Jovanovic, M. Kljajic, A. Petrovic, V. Mizdrakovic, M. Zivkovic, and N. Bacanin, "Modified teaching-learning-based algorithm tuned long short-term memory for household energy consumption forecasting," in *International conference on worldwide computing and its applications*, pp. 347–362, Springer, 1997.

[11] M. Stankovic, L. Jovanovic, M. Antonijevic, A. Bozovic, N. Bacanin, and M. Zivkovic, "Univariate individual household energy forecasting by tuned long short-term memory network," in *Inventive Systems and Control: Proceedings of ICISC 2023*, pp. 403–417, Springer, 2023.

[12] J. Zou, Y. Han, and S.-S. So, *Overview of Artificial Neural Networks*, pp. 14–22. Totowa, NJ: Humana Press, 2009.

[13] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[15] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, pp. 111–132, 2022.

[16] I. Grishchenko, A. Ablavatski, Y. Kartynnik, K. Raveendran, and M. Grundmann, "Attention mesh: High-fidelity face mesh prediction in real-time," *arXiv preprint arXiv:2006.10962*, 2020.

[17] A. V. Tatachar, "Comparative assessment of regression models based on model evaluation metrics," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9, pp. 853–860, 2021.

[18] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.