COMP4801 Final Year Project

Final Report

AI-Powered Attention Monitoring System for Enhancing Online Learning Engagement

Tam Cheung Yan Shojin; 3035927673; BENG(CompSc)

Supervisor: Prof. Zhao Hengshuang

21-4-2025

Abstracts

Online classes have gained popularity since the pandemic, offering greater flexibility and accessibility. However, research indicates that students are more susceptible to distraction during these sessions. This project proposes an innovative application that leverages computer vision AI models, including L2CS-Net for gaze detection, STAR Loss for head pose and facial landmark detection, ResEmoteNet for emotional analysis, and YOLOv11 for real-time phone detection. The application aims to monitor student attentiveness via camera input, enhancing online learning experiences by providing real-time feedback on engagement levels.

Acknowledgement

I would like to express my sincere gratitude to the Department of Computer Science for providing me with the opportunity to undertake this project. My heartfelt thanks go to Prof. Zhao HengShuang for his invaluable advice and guidance throughout the process.

Additionally, I wish to extend my deepest appreciation to Miss Mable Choi for helping me write this report.

Contents

List of Figures

List of Table

Abbreviations & Acronyms

AI - Artificial Intelligence

EAR - Eye aspect ratio

FPS - Frames Per Second

GPU - Graphics Processing Unit

GUI - Graphical User Interface

MSCOCO - Microsoft Common Objects in Context

ONNX - Open Neural Network Exchange

UI - User Interface

YAR - Yawn aspect ratio

Notations & Symbols

S_{Happy} - The percentage of happy emotion predicted by the ResEmoteNet model

x - Attentiveness

$\Sigma\, parameters$ – Sum of all parameters

n – Number of all parameters

1. Solving inattentiveness with AI

Online classes have become essential since the pandemic, enabling students to participate remotely. Educational institutions have increasingly adopted online learning platforms, which allow for greater flexibility and accessibility [1]. However, research indicates that students are more susceptible to distractions during these sessions, adversely affecting their engagement and learning outcomes [2]. The lack of physical presence can lead to a decline in attentiveness [1]. Various studies suggest that students may multitask, engage with their phones [3], or even fall asleep during lectures [4]. This project proposes an innovative application that leverages advanced computer vision AI models to monitor and analyse student attentiveness through camera input, aiming to enhance online learning experiences by providing real-time feedback on student engagement.

1.1. Report outline

The report is structured as follows: Chapter 2 discusses the influential works that inspired this project, providing necessary context and background. Chapter 3 outlines the objectives, clarifying the goals the project aims to achieve. In Chapter 4, the report details the models utilized, including their training processes and the implementation of the program. Chapter 5 presents the results of the training as well as the inference and application of the models. Following this, Chapter 6 highlights the challenges encountered during the project, offering insights into the difficulties faced. Chapter 7 discusses future plans for potential improvements, suggesting directions for future development. Finally, Chapter 8 concludes the report with final insights and reflections, summarizing the overall findings and their implications.

2. Related works

Previous studies have explored various methodologies for monitoring student attentiveness, including:

Head Pose and Facial Landmark Tracking: Analyzing head orientation and facial features to gauge focus. For example, techniques that track head movement can indicate whether a student is looking at the screen [5].

Phone Detection: Identifying the presence of mobile devices to ascertain distractions. Research has shown that phone use during classes correlates with lower academic performance [6].

Eye Tracking: Monitoring eye movements to determine attention levels. Eye tracking can provide insights into where a student's focus lies during a lecture [7].

Student's action detection: Classifying students' behaviours as high attention and low attention behaviour helps determine their attentiveness [8].

These approaches highlight the potential for integrating multiple data sources to create a comprehensive attentiveness monitoring system. Inspired by these studies, the proposed application will leverage open-source AI models to perform the four aforementioned tasks for assessing students' attentiveness.

3. Objectives

The proposed application will be built on the OpenCV framework for real-time computer vision. It aims to integrate multiple AI models:

1. L2CS-Net [9]: A gaze detection model to assess where the student is looking.
2. STAR Loss [10]: For detecting head orientation and facial landmarks.
3. ResEmoteNet [11]: To analyse the student's emotional state.
4. YOLOv11 [12]: For real-time detection of phones and monitoring of student actions.

These models will be trained on diverse datasets to enhance their predictive capabilities. The outputs will be synthesized to calculate a comprehensive attentiveness score for each student. A user-friendly interface will display these results alongside insights derived from the model predictions.

4. Methodology

The following section covers the AI models utilised in the project, the calculation of attentiveness, and the implementation of the application. Chapters 4.1, 4.2, 4.3, and 4.4 present the outputs and attentiveness calculations for L2CS-Net, STAR Loss model, ResEmoteNet, and YOLOv11, respectively. Chapter 4.5 provides an overview of the overall attentiveness calculation. Chapter 4.6 details the datasets used for training. Finally, Chapters 4.7 and 4.8 discuss the implementation of training and deployment code.

4.1. L2CS-Net

L2CS-Net predicts the gaze direction of a student [9]. Attentiveness will be assumed if the student is looking forward; while inattentiveness will be assumed if the student is looking sideways, upward, downward, or backwards.

4.2. STAR Loss

STAR Loss detects head pose and facial landmarks [10]. For the facial landmarks, the model predicts 68 facial landmarks as proposed by the 300W database [13], which is shown in figure 1. Eye aspect ratio (EAR) and yawn aspect ratio (YAR) will be calculated, which are indicators of the drowsiness of the person, and will be used to calculate attentiveness [5].

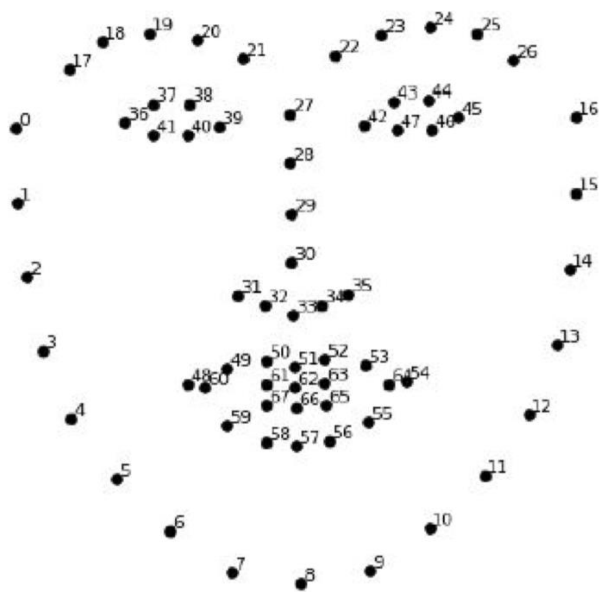

Figure 1: 68 Facial Landmarks [14]

The calculation of EAR is as follows [5]:

$$EAR = \frac{EAR1 + EAR2}{2} \tag{1}$$

$$EAR1 = \frac{||37 - 41|| + ||38 - 40||}{2||36 - 39||} \tag{2}$$

$$EAR2 = \frac{||43 - 47|| + ||44 - 46||}{2||42 - 45||} \tag{3}$$

The calculation of YAR is as follows [5]:

$$YAR = \frac{||61 - 67|| + ||62 - 66|| + ||63 - 65||}{2||64 - 60||} \tag{4}$$

EAR1 and EAR2 are the eye aspect ratios of the left and right eyes, respectively. For each landmark, a value will be calculated in (x-y) format, using the coordinates of the landmark [15]. The numbers in the formulas, such as 37 and 41, refer to the values correlated to the 37^{th} and 41^{st} landmarks, respectively.

As for the head pose, similarly, for gaze direction, attentiveness will be determined by the direction of the head pose.

4.3. ResEmoteNet

ResEmoteNet predicts the emotions of the student [11]. The model categorises emotions as one of the seven classes: angry, disgusted, fearful, happy, neutral, sad, and surprised. Attentiveness will be calculated based on the affective model shown in Figure 1 [16].

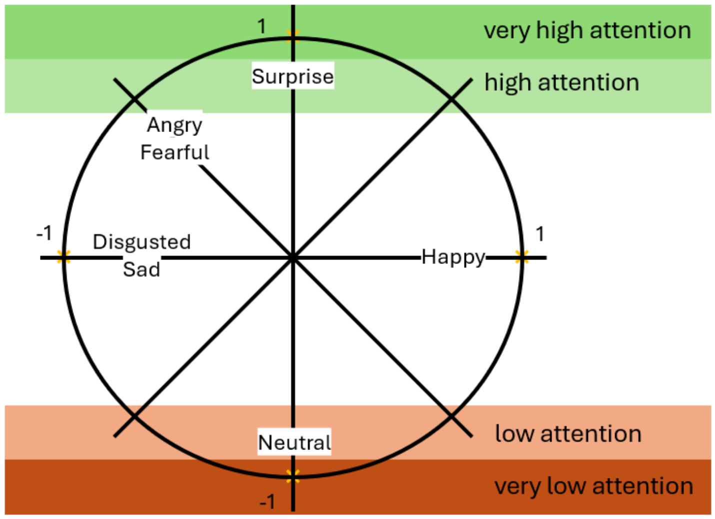

Figure 2: Affective Model modified from [16] to match with classes of ResEmoteNet

We calculate the attentiveness as follows [17]:

$$S_1 = S_{Happy} + (-1) \cdot (S_{Sad} + S_{Disgusted}) \tag{5}$$

$$S_2 = S_{Surprise} + (-1) \cdot S_{Neutral} \tag{6}$$

$$S_3 = S_{Angry} + S_{Fearful} \tag{7}$$

Where $S = \{S_{Happy}, S_{Sad}, S_{surprise}, S_{Neutral}, S_{Disgusted}, S_{Fearful}\}$ are the percentages of each of the emotions predicted by the model. The algorithm then calculates attentiveness x:

$$x = \sqrt{\left(\overline{S_1} + \frac{\overline{S_3}}{\sqrt{2}}\right)^2 + \left(\overline{S_2} + \frac{\overline{S_3}}{\sqrt{2}}\right)^2} \cdot \frac{\overline{S_2} + \frac{\overline{S_3}}{\sqrt{2}}}{\left|\overline{S_2} + \frac{\overline{S_3}}{\sqrt{2}}\right|} + 1 \tag{8}$$

Here, $\sqrt{\left(\overline{S_1} + \frac{\overline{S_3}}{\sqrt{2}}\right)^2 + \left(\overline{S_2} + \frac{\overline{S_3}}{\sqrt{2}}\right)^2}$ indicates the overall emotional intensity, while $\frac{\overline{S_2} + \frac{\overline{S_3}}{\sqrt{2}}}{\left|\overline{S_2} + \frac{\overline{S_3}}{\sqrt{2}}\right|}$ serves as the dimensional factor to assess whether a student is focused or distracted. The offset 1 is added so the range of the attentiveness becomes [-1,1].

4.4. Yolov11

Yolov11 is a detection model suitable for real-time detection [12]. It is selected due to its performance. The base model will be used to detect mobile phones, and it is trained with MSCOCO [18]. Inattentiveness will be assumed if a mobile phone is detected.

4.5. Overall attentiveness calculation

The overall attentiveness is the mean of all the parameters calculated with the above models, as suggested in [7].

$$Attentiveness = \frac{\Sigma\, parameters}{n} * 100 \tag{9}$$

4.6. Dataset preparation

Most AI models will be trained using the datasets that were used to develop the pretrained models provided by their authors. Details of the datasets are presented in Tables 4.1 and 4.2.

Table 4.1: Datasets used for training STAR Loss [19]

Name	Published	#Marks	#Samples
AFW [20]	2013	68	337
HELEN [21]	2013	68	2330
IBUG [22]	2013	68	135
LFPW [23]	2013	68	1035
Total			**3837**

Table 4.2: Datasets used for training L2CS-Net, ResEmoteNet, and YOLOv8

AI model	Task	Dataset	#Classes	Dataset size
L2CS-Net	Gaze detection	MPIIGaze [24]	1	15 subjects, 213659 images
ResEmoteNet	Emotion detection	FER2013 [25]	7	34034 images

YOLOv8	Phone detection	MSCOCO [18]	80	330000 images, 11000 instances of cell phone

4.7. Model training

All models will be trained from scratch using the prepared datasets, except for the phone detection model. Due to the size of the MSCOCO dataset, we will utilize a pretrained model for this task instead.

4.8. Program coding

The training and deployment code are implemented using Python within the Opencv framework, which is known for its efficiency and flexibility in handling computer vision tasks. Python will be used for ease of programming.

4.9. Summary

This chapter outlines the rationale behind our project. We introduced the following AI models: L2CS-Net, the STAR Loss model, the ResEmoteNet, and YOLOv11. We also defined the methods for calculating attentiveness, dataset preparation, and the framework that will be used for the program. The next chapter will discuss our current progress.

5. Experiments and Results

5.1. Overview

This chapter outlines the current progress of the project. Chapter 5.2 provides details on model training. Chapter 5.3 presents the results of the training, while Chapters 5.4 and 5.5 showcase the outcomes of real-time inference of models and the application.

5.2. Model training

L2CS-Net, ResEmoteNet, and STAR Loss were trained using the official training code provided in their respective GitHub repositories. Table 5.1 shows the configuration of the training.

Table 5.1: Configuration of the training

AI model	Batch size	#Epochs	Learning rate
L2CS-Net	16	50	0.00001
ResEmoteNet	16	80	0.001
STAR Loss	32	500	0.001

5.3. Training results

Testing for L2CS-Net, ResEmoteNet, and STAR Loss was conducted using the testing sets of the datasets. The training results of the models are shown in Table 5.2.

Table 5.2: Training results of L2CS-Net, ResEmoteNet, and STAR Loss

AI model	Name of Metric	Value
L2CS-Net	Gaze Angular Error	2.307
ResEmoteNet	Accuracy	0.900
STAR Loss	Normalized Mean Error	0.028643

While the accuracy is satisfactory, it indicates that incorporating additional datasets could help achieve higher performance.

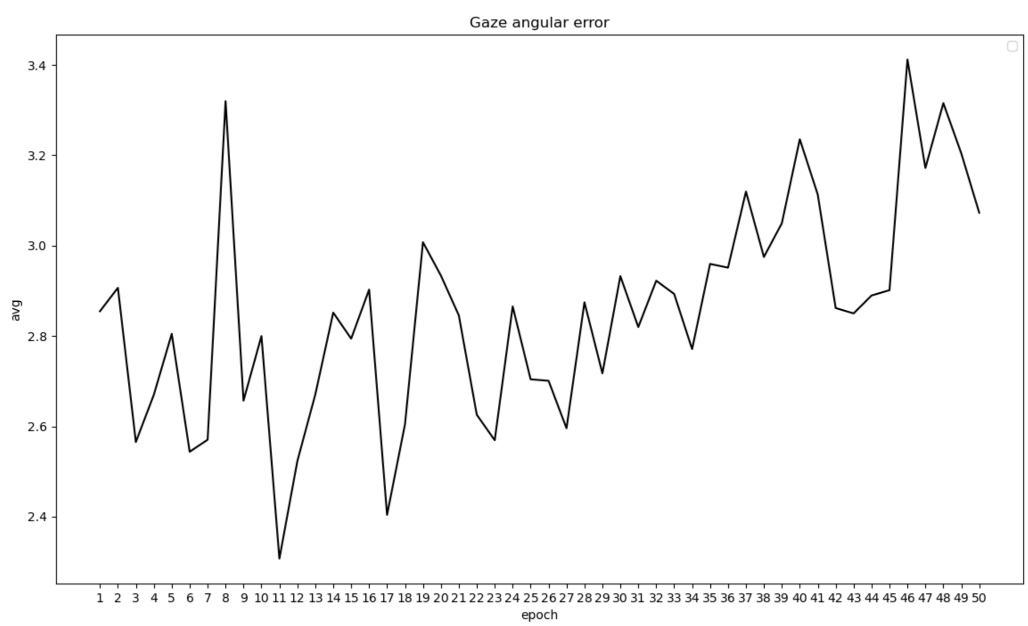

Figure 3: Gaze Angular Error of L2CS-Net

For L2CS-Net, epoch 11 yielded the best results, as illustrated in Figure 3. Throughout training, the error value exhibited fluctuations, notably rising after epoch 11, indicating the onset of overfitting. To address this issue, adjustments such as tuning the learning rate or increasing the dataset size could be beneficial.

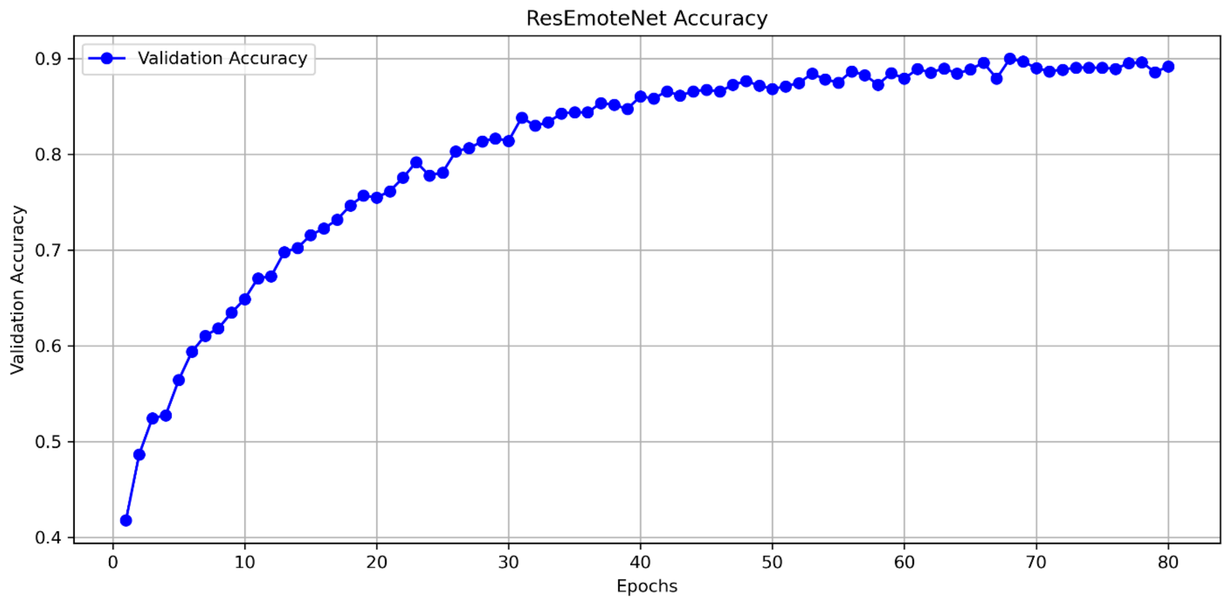

Figure 4: Accuracy of ResEmoteNet

Figure 5: Training Loss of ResEmoteNet

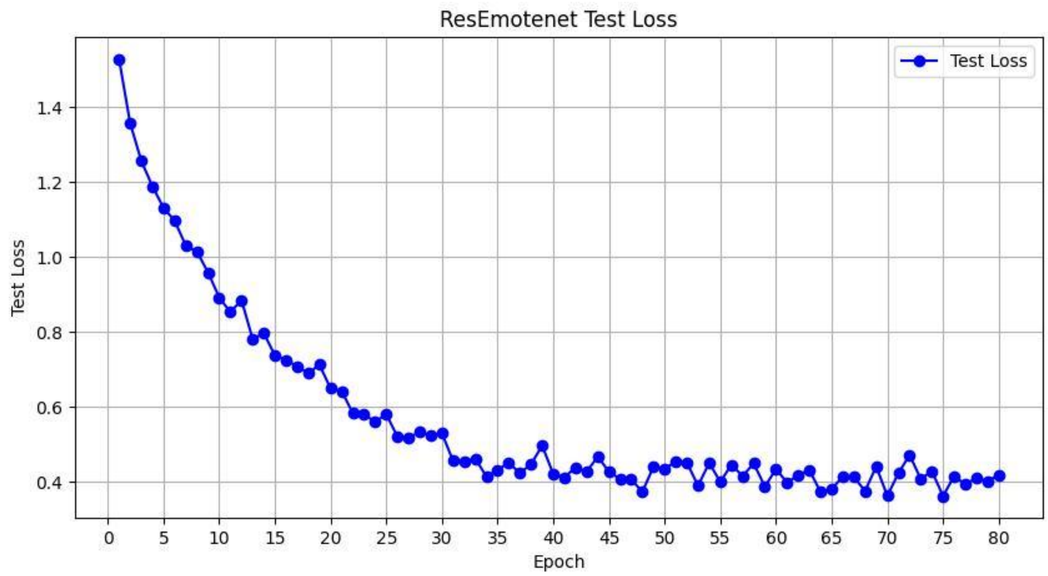

Figure 6: Testing Loss of ResEmoteNet

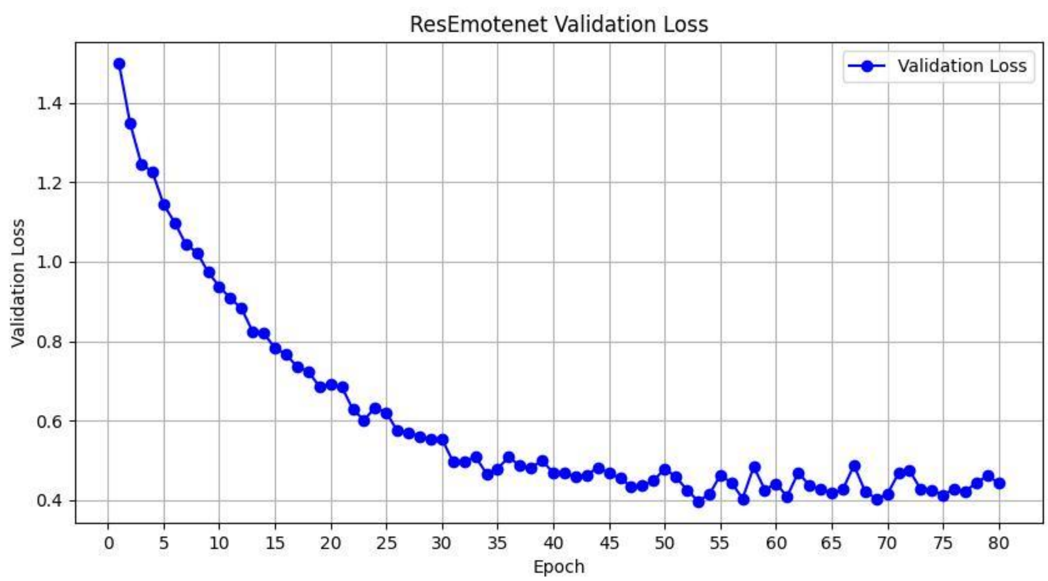

Figure 7: Validation Loss of ResEmoteNet

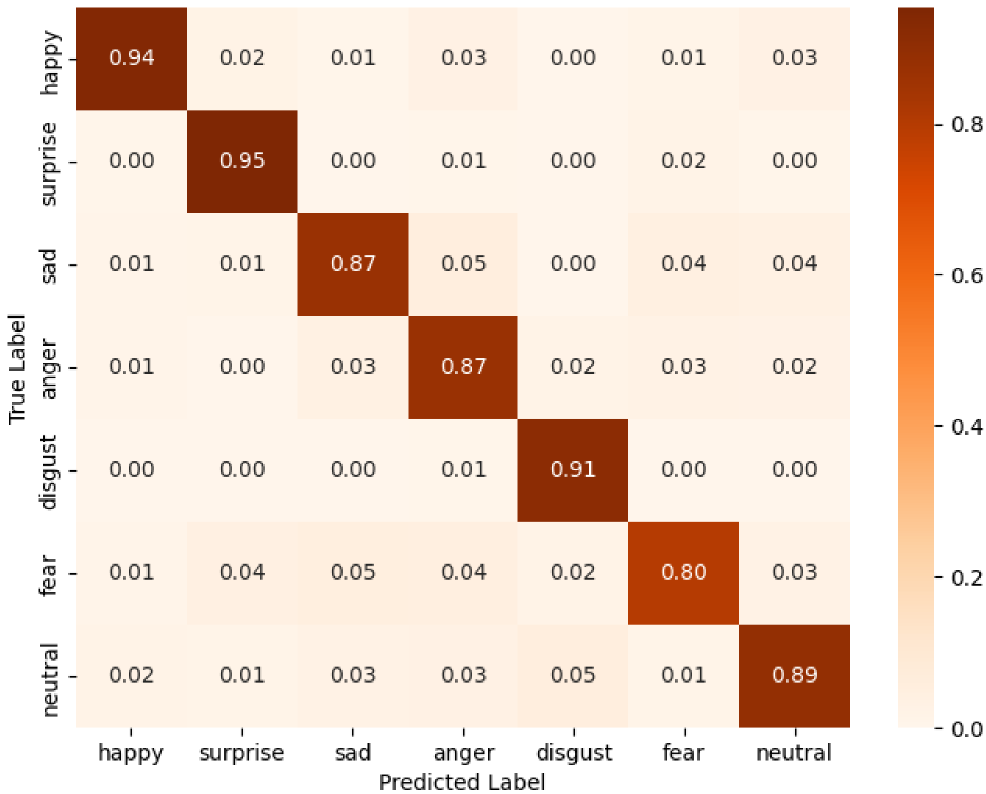

Figure 8: Confusion Matrix of ResEmoteNet

In contrast, ResEmoteNet demonstrated a stable learning curve, as depicted in Figures 4, 5, 6, and 7. This stability suggests that the current configurations are well-optimised, effectively avoiding overfitting issues. Figure 8, which is a confusion matrix, shows that the model effectively captures the distinctions among the classes.

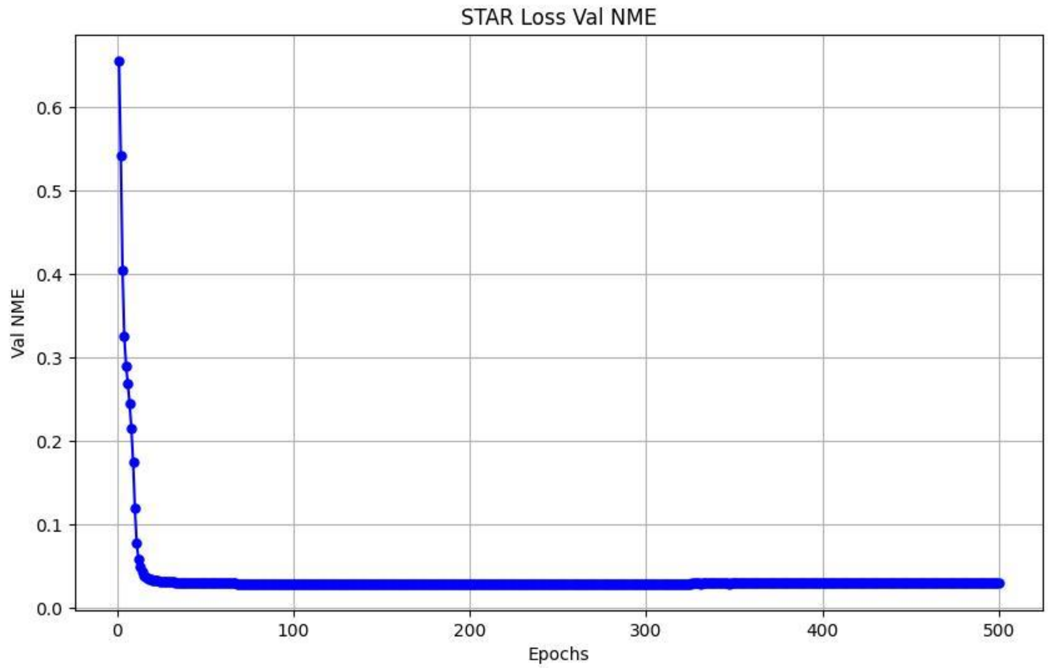

Figure 9: STAR Loss Validation Normalized Mean Error

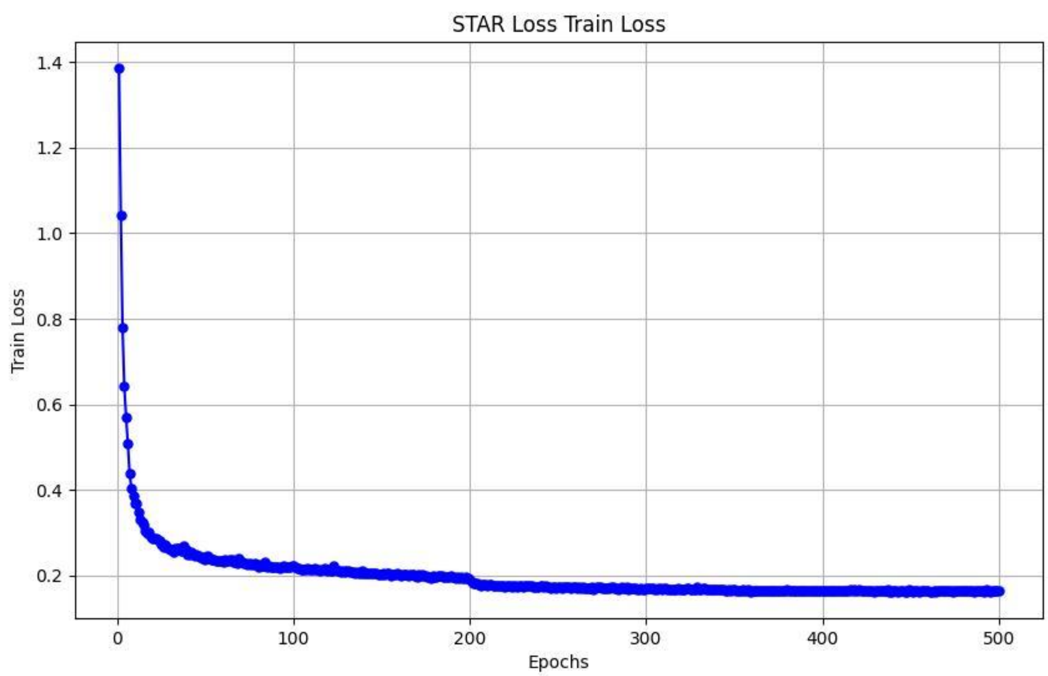

Figure 10: STAR Loss Training Loss

Similarly, the STAR Loss exhibits a smooth learning curve, as illustrated in Figures 9 and 10. This behaviour suggests that the model is well-configured, reflecting stable training and effective convergence. The consistent performance throughout the learning process indicates that the configurations employed are beneficial for achieving optimal results.

5.4. Real-time inference of models

Real-time inference was conducted to evaluate the speed of the AI models. The frames per second (FPS) for each model were recorded in Table 5.3.

Table 5.3: Frames per second of AI models

AI model	Average FPS
L2CS-Net	9.2
ResEmoteNet	25.3
STAR Loss	6.3

These results suggest that performance improvements are necessary, as the speeds are relatively slow.

5.5. Application inference

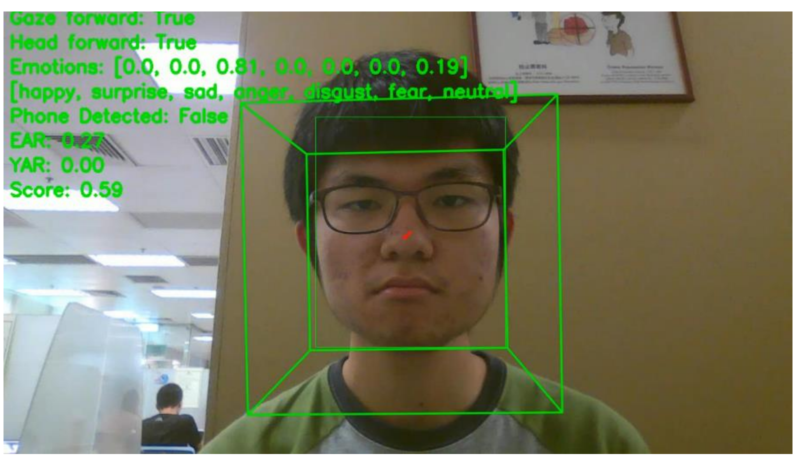

Figure 11: A user with head and gaze directed forward

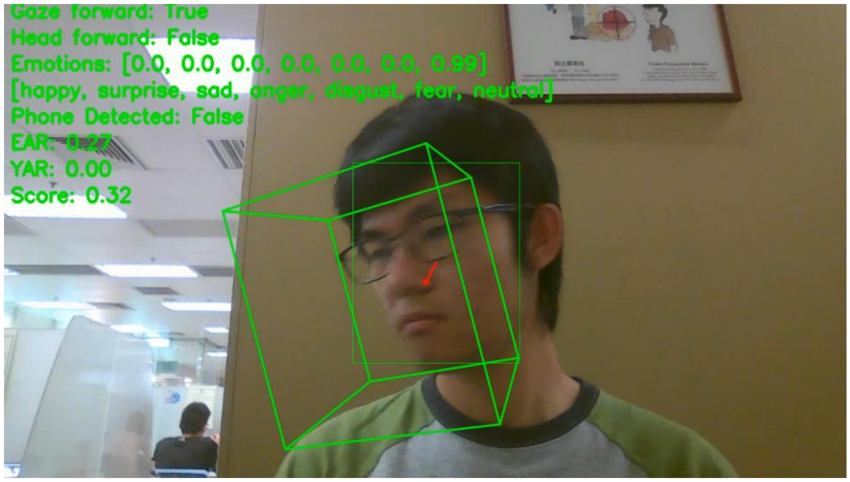

Figure 12: A user with head facing sideways

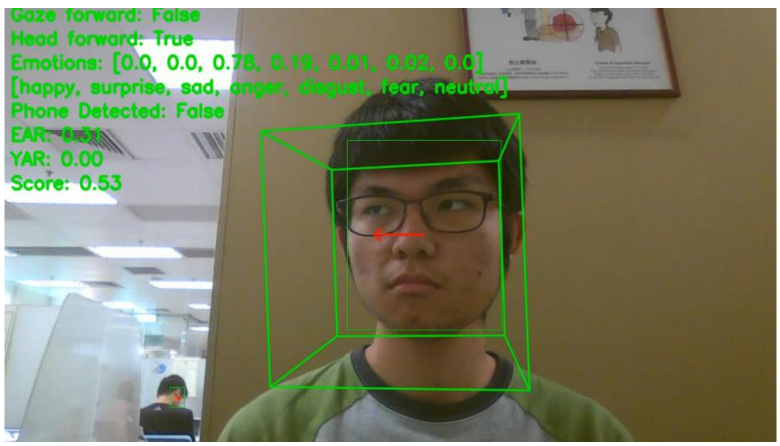

Figure 13: A user with gaze directed sideways

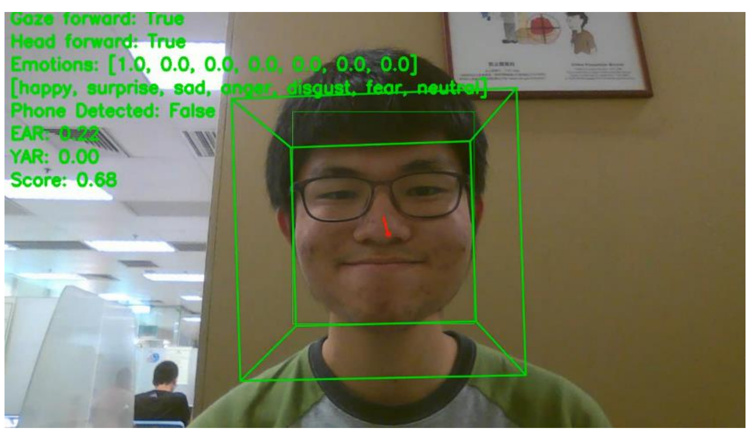

Figure 14: A user displaying a happy expression

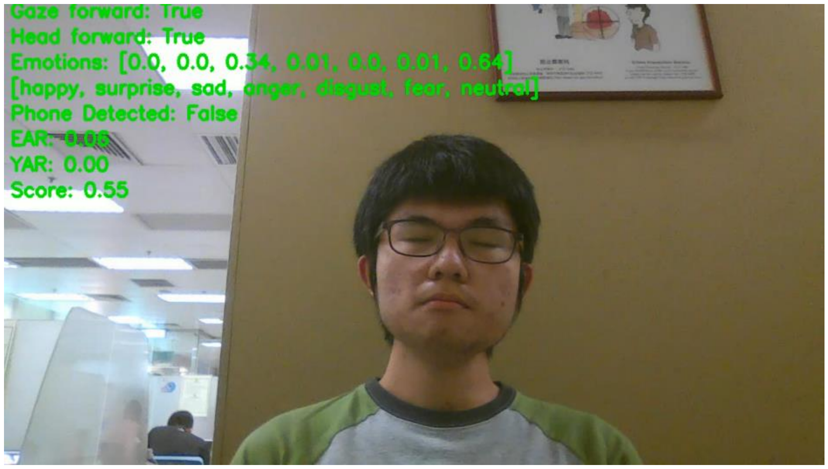

Figure 15: A user with eyes shut

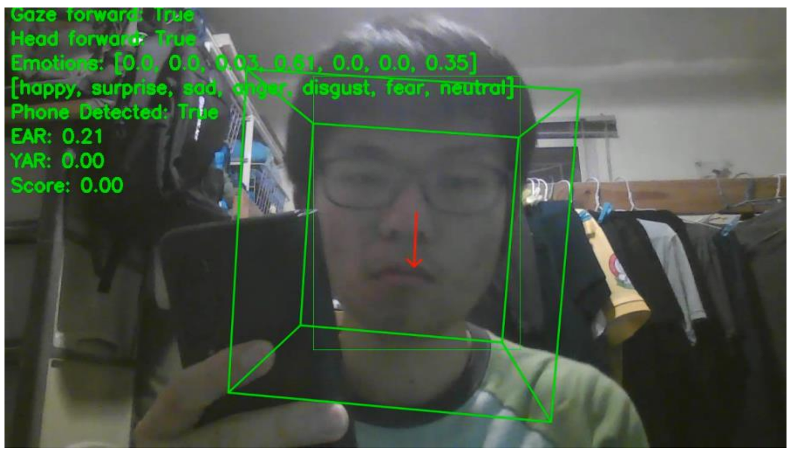

Figure 16: A user with a phone

While we do not have a dedicated dataset for testing against ground truth, initial inferences suggest that the parameters generated by the model predictions are generally accurate. For instance, in Figure 12, the attentiveness score is low when the user's head is not facing forward, which aligns with expectations. However, there are instances where the scores are unexpected; in Figure 15, the score remains relatively high at 0.55 despite the user having closed eyes.

Additionally, the average frames per second (FPS) recorded is 1.4, which is not practical for real-time application. This low performance necessitates further optimization to ensure that the system can operate effectively in real-time scenarios. Future improvements will focus on enhancing processing speed and refining the accuracy of attentiveness assessments.

6. Challenges

In addition to the issues discussed in Section 5, a significant challenge is the limited size of the datasets used for all three models. The lack of sufficiently large datasets restricts the models' ability to perform accurately in real-life, uncontrolled environments. This limitation affects their effectiveness when applied in practical scenarios, where variability in user behavior and conditions is expected.

7. Future plans

The models have significant potential for enhancement through the integration of additional datasets. Specifically, L2CS-Net can be further trained using the Gaze360 [26] dataset, while STAR Loss can benefit from training on diverse datasets such as 300-W [22], 300-VW [27], and AFLW2000-3D [28]. This expansion of training data will help improve the accuracy and robustness of the models.

In addition to model training, the current application can be enhanced by incorporating a Graphical User Interface (GUI) using Qt. This improvement is expected to significantly enhance user experience, making the application more intuitive and user-friendly.

To refine the attentiveness score calculation, we may create a dataset with attentiveness labels. This will allow for the development of a more effective formula for deriving the score, thereby increasing its accuracy.

Finally, performance optimization is a crucial aspect of future development. Converting the models to ONNX format could lead to improved performance and better compatibility with various platforms. Currently, the application utilizes inference pipelines from three models hosted on GitHub, each employing different face detectors (dlib and RetinaFace). Standardizing on a single face detector across all models could streamline processes and enhance overall performance. By implementing these strategies, we can optimize both the functionality and user interaction of the application.

8. Conclusion

This project addresses the significant challenge of student distraction during online classes by developing an AI-powered attention monitoring system. Utilizing advanced computer vision models—L2CS-Net for gaze detection, STAR Loss for head pose analysis, ResEmoteNet for emotional assessment, and YOLOv11 for phone detection—we aim to provide real-time insights into student engagement.

The application is designed to track various indicators of attentiveness and deliver actionable feedback to enhance the online learning experience. By focusing on these metrics, we hope to improve student focus and participation in remote learning environments.

The application can be improved by refining model accuracy through additional training datasets, optimizing performance for real-time processing, and enhancing the user interface for a more intuitive experience. Ultimately, we hope this system will boost engagement in online education.

References

[1] C. B. Hodges, S. Moore, B. B. Lockee, T. Trust and M. A. Bond, "The difference between emergency remote teaching and online learning," 2020.

[2] K. A. Aivaz and D. Teodorescu, "College Students' Distractions from Learning Caused by Multitasking in Online vs. Face-to-Face Classes: A Case Study at a Public University in Romania," *Int J Environ Res Public Health,* vol. 18, 2022.

[3] R. Junco, "In-class multitasking and academic performance," *Computers & Education,* vol. 59, no. 2, pp. 505-514, 2012.

[4] A. J. Mendez, "The effects of online learning on student engagement: A study on sleepiness," *Journal of Online Learning Research,* vol. 6, no. 1, pp. 45-63, 2020.

[5] M. Elbawab and R. Henriques, "Machine Learning applied to student attentiveness detection: Using emotional and non-emotional measures," *Educ Inf Technol,* vol. 28, p. 15717–15737, 2023.

[6] M. K. Hossen and M. S. Uddin, "Attention monitoring of students during online classes using XGBoost classifier," *Computers and Education: Artificial Intelligence,* vol. 5, 2023.

[7] K. Vignesh, V. D. N. Gupta and H. Kumar, "Attentiveness Recognition System for Online Classes using OpenCV," in *2022 2nd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2022.

[8] Z. Trabelsi, F. Alnajjar, M. M. A. Parambil, M. Gochoo and L. Ali, "Real-Time Attention Monitoring System for Classroom: A Deep Learning Approach for Student's Behavior Recognition," *Big Data and Cognitive Computing,* vol. 7, no. 1, p. 48, 2023.

[9] A. A. Abdelrahman, T. Hempel, A. Khalifa, A. Al-Hamadi and L. Dinges, "L2cs-net: Fine-grained gaze estimation in unconstrained environments," in *2023 8th International Conference on Frontiers of Signal Processing (ICFSP)*, 2023.

[10] Z. Zhou, H. Li, H. Liu, N. Wang, G. Yu and R. Ji, "STAR Loss: Reducing Semantic Ambiguity in Facial Landmark Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[11] A. K. Roy, H. K. Kathania, A. Sharma, A. Dey and M. S. A. Ansari, "ResEmoteNet: bridging accuracy and loss reduction in facial emotion recognition," 2024.

[12] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," 2024.

[13] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou and M. Pantic, "300 Faces In-The-Wild Challenge: database and results," *Image and Vision Computing,* vol. 47, pp. 3-18, 2016.

[14] P. Korshunov and S. Marcel, "Speaker inconsistency detection in tampered video," in *2018 26th European signal processing conference (EUSIPCO)*, Rome, Italy, 2018.

[15] N. A. Shah, K. Meenakshi, A. Agarwal and S. Sivasubramanian, "Assessment of Student Attentiveness to E-Learning by Monitoring Behavioural Elements," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2021.

[16] K. Altuwairqi, S. K. Jarraya, A. Allinjawi and M. Hammam, "A new emotion–based affective model to detect student's engagement," *Journal of King Saud University - Computer and Information Sciences,* vol. 33, no. 1, pp. 99-109, 2021.

[17] Q. T. Le, T. K. Do, C. B. Ngo, T. T. T. Dang, T. M. T. Luong, D. C. Duong, H. M. H. Nguyen and H. T. Nguyen, "Monitoring and improving student attention using deep learning and wireless sensor networks," *Sensors and Actuators A: Physical,* vol. 367, 2024.

[18] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick and P. Dollár, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference*, Zurich, Switzerland, 2014.

[19] G. Ying, "facial-landmark-dataset," [Online]. Available: https://github.com/yinguobing/facial-landmark-dataset. [Accessed 26 January 2025].

[20] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *CVPR*, 2012.

[21] V. Le, J. Brandt, Z. Lin, L. Bourdev and T. S. Huang, "Interactive facial feature localization," in *ECCV*, 2012.

[22] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *CVPR*, 2013.

[23] P. Belhumeur, D. Jacobs, D. Kriegman and N. Kumar, "Localizing parts of faces using a consensus of exemplars," in *CVPR*, 2011.

[24] X. Zhang, Y. Sugano, M. Fritz and A. Bulling, "MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 41, no. 1, pp. 162-175, 2017.

[25] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, et al., "Challenges in representation learning: A report on three machine learning contests," in *INCONIP*, 2013.

[26] P. Kellnhofer, A. Recasens, S. Stent, W. Matusik and A. Torralba, "Gaze360: Physically unconstrained gaze estimation in the wild," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[27] J. Shen, S. Zafeirious, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos and M. Pantic, "The first facial landmark tracking in-the-wild challenge: Benchmark and results," in *ICCVW*, 2015.

[28] X. Zhu, Z. Lei, X. Liu, H. Shi and S. Z. Li, "Face alignment across large poses: A 3d solution," in *CVPR*, 2016.

[29] J. Fredricks, P. C. Blumenfeld and A. Paris, "School engagement: Potential of the concept, state of the evidence.," *Review of Educational Research,* vol. 74, no. 1, pp. 59-109, 2004.

[30] V. Sharma and M. Gupta, "Classroom-Monitoring-Action-Dataset," [Online]. Available: https://github.com/vijetait/Classroom-Monitoring-Action-Dataset?tab=readme-ov-file. [Accessed 30 September 2024].

[31] G. Ying, "facial-landmark-detection-hrnet," [Online]. Available: https://github.com/yinguobing/facial-landmark-detection-hrnet. [Accessed 25 January 2025].