Head Pose and Eye Blink-Based Mouse Control

Zihao Huang
Department of Computer Science and Software Engineering
University of Canterbury
Canterbury, New Zealand
Email: joe.huang@lincolnuni.ac.nz

Richard Green
Department of Computer Science and Software Engineering
University of Canterbury
Canterbury, New Zealand
Email: richard.green@canterbury.ac.nz

***Abstract*—This paper introduces an innovative approach to computer interaction, employing a combination of techniques for monitoring and analyzing facial movements. Our method utilizes a model that incorporates cascaded regression and Convolutional Neural Networks (CNN) to accurately project a 3D model onto the human face. This facilitates precise identification of facial landmarks and measurement of yaw and pitch angles, allowing us to interpret facial expressions and movements with precision, even when the face is partially invisible in the camera.**

The central idea of our approach revolves around the integration of head pose estimation and eye blink detection through the utilization of facial landmarks, head pose angles, and a CNN model. This integration enables users to control the mouse cursor and perform clicks by purposefully moving their head and blinking, providing an intuitive alternative input method. This approach is especially advantageous for individuals with motor disabilities, contributing to a more inclusive computer interaction experience.

Our experiments confirm the effectiveness of our approach, showcasing reliable control of the mouse cursor and achieving a 96% accuracy rate in the MRL Eye Dataset. To ensure precise click controls, we employ thresholding and timing control techniques. In conclusion, our method, which combines advanced facial landmark analysis, neural network capabilities, and practical applications, emerges as a promising solution in the realm of human-computer interaction, enhancing the computing experience for a diverse user base.

Keywords:* *Mouse control, Head pose estimation, Facial landmarks, 3D dense face alignment, 3DDFA, Eye blink detection, CNN

I. Introduction

In the ever-evolving landscape of human-computer interaction (HCI), accessibility and user experience are at the forefront of research and innovation. Traditional input devices, though widely used, may present challenges for individuals with motor disabilities. Addressing this concern, our paper introduces a novel approach that leverages cutting-edge computer vision techniques, specifically facil landmarks, head pose estimation and eye blink detection. By incorporating a sophisticated Convolutional Neural Network (CNN) in conjunction with 3DDFA [1], the proposed approach aims to innovate HCI paradigms by enabling mouse control without the need for physical input devices.

II. Background

A. Mouse control

Previous studies exhibit diversity in terms of technologies employed. While some research projects utilize specific hardware to gather head pose information [2][3], an increasing number of studies leverage cameras due to their widespread availability in laptops and smart devices. This shift eliminates the need for extensive time investment in hardware design and prototyping, allowing researchers to concentrate more on refining software algorithms. The subsequent section provides an overview of related works conducted in this field.

A hardware-based implementation of a head-wearable mouse was introduced, employing a straightforward, cost-effective infrared mechanism [2]. This approach utilized two infrared (IR) LEDs to gauge the distance between the user's chin and the sensor deck. Employing uncomplicated algorithms, the approach converted distance data into X and Y coordinates on the screen, communicating this information to the host machine via Bluetooth Low Energy technology. Positioned beneath the user's chin, the two infrared LED sensors accurately identified yaw and pitch movements, generating distinct signals in Figure 1. The device offered both direct map mode and joystick mode functionalities. In direct map mode, users initialized minimum and maximum head movements, mapping them to the screen's X and Y coordinates. Alternatively, joystick mode subtly adjusted the mouse cursor by one pixel upon detecting head movement. However, the solution exhibited drawbacks, particularly in stability under varying light conditions due to its reliance on infrared technology. Additionally, addressing the unwelcome appearance issue necessitated a sophisticated design.

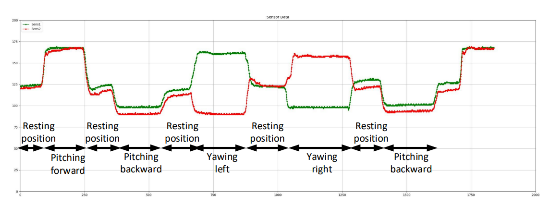


Fig. 1. Sensor signal

Marker-based approaches have been explored in various studies, such as those discussed in [4] and [5]. In the former, researchers employed a sports-type tape adorned with a printed blue circle worn on the head. The methodology involved capturing frames frame by frame with a camera to detect the position of the blue circle. The method identified the center, as well as the up, down, left, and right positions based on the calculated relative position of the blue circle in relation to the image center.

Another marker-based study, detailed in [5], utilized four markers strategically placed on the user's face, forming the corners of a square-shaped token. The approach focused on detecting and tracking these four markers using a calibrated monocular camera. The algorithm employed in this study estimated a 3D rotation matrix and translation by analyzing the point and line correspondences between the 2D head pose coordinates and the 3D camera coordinates (1). Xc, Yc, and Zc represent the 3D camera coordinates, while Xh, Yh, and Zh represent the 2D head pose coordinates. The rotation and translation matrix is represented by (2).

$$\begin{bmatrix} Xc \\ Yc \\ Zc \\ 1 \end{bmatrix} = \mathbf{T_{CH}} \cdot \begin{bmatrix} Xh \\ Yh \\ Zh \\ 1 \end{bmatrix} \tag{1}$$

$$\mathbf{T_{CH}} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Upon obtaining the rotation and translation matrix, the mouse's X and Y coordinates can be calculated with respect to the screen width and height. This study demonstrated commendable results in mouse cursor control, exhibiting lower errors compared to those controlled by nose tracking. Nevertheless, it still necessitated the use of markers.

In more recent investigations, there has been a trend towards utilizing machine learning to achieve more accurate head pose estimation [6] [7]. Some studies directly produce yaw, pitch, and roll angles from machine learning networks, while others employ networks to first estimate and align facial landmarks. Subsequently, these studies calculate the yaw, pitch, and roll angles by solving for the rotation and translation between 2D and 3D coordinate systems based on the aligned facial landmarks. Our proposed method, leveraging the 3DDFA [1] [8] [9], aligns with the latter approach, emphasizing head pose estimation through facial landmark analysis and subsequent geometric computations.

B. Eye blink detection

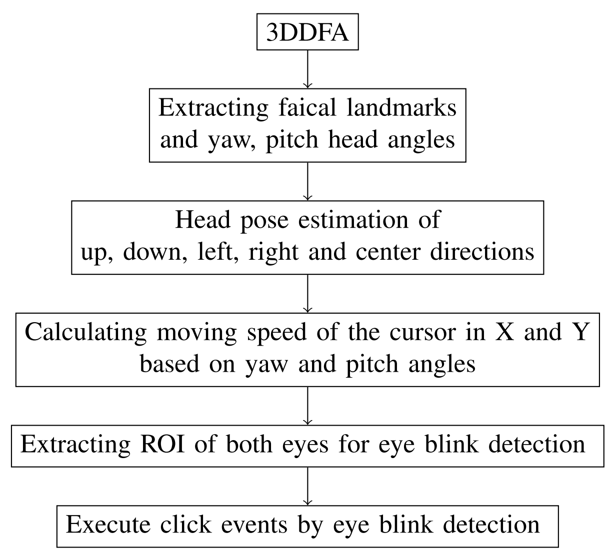
In a previous study [10], an eye region's Haar-like feature detector was employed, followed by feeding the data into a template matching-based method for eye tracking and blink detection. The detected eyes were tracked using a normalized cross-correlation method. While this approach demonstrated accurate eye blink detection, its applicability is limited to specific users, as it necessitates a template image from each individual during the software setup process.

In the study [11] [12] [13], a common approach involves employing a cascade of regressors to pinpoint the facial 68 landmarks [14]. Specifically, landmarks 27 (center of the eye), 36 (left edge of the left eye), and 45 (right edge of the right eye) are extracted to calculate the eye aspect ratio (EAR) for discerning whether the eyes are open or closed. However, relying on the EAR method for accurate eye blink detection can be challenging, as it hinges on the accuracy of the facial landmarks model. Many facial landmarks models struggle to precisely locate landmarks in the eye area, making it necessary to undertake further research for obtaining accurate eye landmarks [11], especially when faces exhibit different expressions such as smiling or yawning [12].

In the investigation documented in [15] [16], researchers developed a dedicated model specifically designed to detect the open or closed state of the eye. The findings indicate that the model is proficient in accurately discerning the eye state, enabling applications such as triggering a click event with a brief eye closure of 0.5 seconds and a double-click event with a closure duration of 1 second. Our proposed method similarly adopts this effective approach for eye blink detection.

III. PROPOSED METHOD

Our proposed method is base on facial landmarks and yaw, pitch head angles to control the mouse cursor. The proposed method is comprised of three main components: head pose estimation, eye blink detection, and mouse control. The following sections provide a detailed overview of each component.



A. Facial landmarks and head pose angles

The core of our proposed method centers on utilizing facial landmarks and head pose angles, and the accuracy of these estimations is of utmost importance, especially in applications like mouse control. In this context, the 3DDFA (3D Dense Face Alignment) model stands out as a robust solution for retrieving 68 facial landmarks, in addition to the critical yaw and pitch angles essential for outlining facial orientation in three-dimensional space. The 3DDFA model employs a 3D morphable model fitted with cascaded CNN, showcasing its expertise in generating accurate and detailed facial landmark predictions.

Distinguishing itself from other facial landmarks detectors, such as the one referenced [14], 3DDFA overcomes the challenge of detecting facial landmarks at large angles, where certain landmarks might be obscured from the frame. Noteworthy is 3DDFA's proven capability to operate effectively at angles up to 90 degrees of yaw and pitch, showcasing its prowess in handling extensive facial orientations. This broad detection capability offers tangible benefits to our proposed method, enhancing its usability by eliminating the necessity for users to position their entire faces within the camera frame for mouse control. The 3DDFA model's ability to function seamlessly at large angles represents a significant advancement, streamlining user interaction and making our proposed method more accessible and user-friendly.

B. Head pose directions

Head pose estimation involves categorizing directions into up, down, left, right, and center orientations. The approach adopted in this method hinges on a comparison between the nose tip landmark and the centroid of the 68 facial landmarks, which is computed using the formula given in Equation (3). Specifically, the nose tip landmark is identified as the 30th landmark, serving as a key reference point for determining the overall head pose. This methodology provides a robust framework for discerning and classifying head orientations based on the spatial relationships among facial landmarks. Algorithm 1 is employed for discerning the head pose direction, with the crucial parameter "threshold" playing a defining role in this determination. The threshold serves as a constant value utilized to ascertain the head pose direction. Specifically, it is set to 8 for the up, left, and right directions, and 2 for the down direction. The choice of these threshold values is experimentally determined, ensuring optimal performance and adaptability to diverse scenarios.

$$
\begin{aligned}
C_x &= \frac{1}{N}\sum_{i=1}^{N} x_i \\
C_y &= \frac{1}{N}\sum_{i=1}^{N} y_i
\end{aligned} \tag{3}
$$

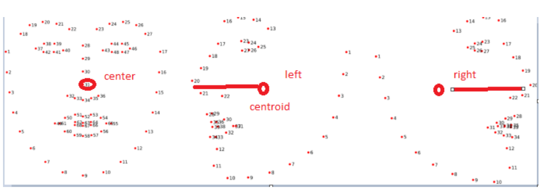


Fig. 2. Directions

Algorithm 1 Calculate Movement Direction

$x, y, threshold \leftarrow$ nose_landmark$[0]$, nose_landmark$[1]$, 8
if $x <$ face_centroid$[0]$ $-$ threshold **then**
  return "left"
else if $x >$ face_centroid$[0]$ $+$ threshold **then**
  return "right"
else if $y <$ face_centroid$[1]$ $-$ threshold **then**
  return "up"
else if $y >$ face_centroid$[1] + 2$ **then**
  return "down"
else
  return "center"
end if

C. Mouse cursor control

After determining the movement direction, the velocity becomes closely tied to the yaw and pitch angles in the X and Y directions, respectively. Figure 3 vividly illustrates the exponential increase in acceleration as both yaw and pitch angles elevate. The computational essence of this acceleration is expressed by the formula in Equation (4). Consequently, this calculated speed factor guides the mouse cursor seamlessly in accordance with the identified direction.

Operating in a joystick-like mode, the mouse cursor incorporates an advanced speed control mechanism. This feature is essential for smooth navigation on large screens, providing precise control for detailed operations. The speed control integration not only enables fluid cursor movements over greater distances but also enhances precision for intricate tasks. The code executes mouse cursor movements every 200 milliseconds to improve accuracy. Additionally, an active mode is applied for mouse control. If the mouse cursor remains in an area for 10 seconds, it becomes locked. Users need to shake their head to unlock it.

$$
\begin{aligned}
\text{speed\_x} &= \text{yaw} \cdot \exp(|\text{yaw}| \cdot 0.04) \\
\text{speed\_y} &= \text{pitch} \cdot \exp(|\text{pitch}| \cdot 0{,}04) \\
\text{move\_x} &= \text{yaw} \cdot \text{speed\_yaw} \\
\text{move\_y} &= \text{pitch} \cdot \text{speed\_pitch}
\end{aligned} \tag{4}
$$

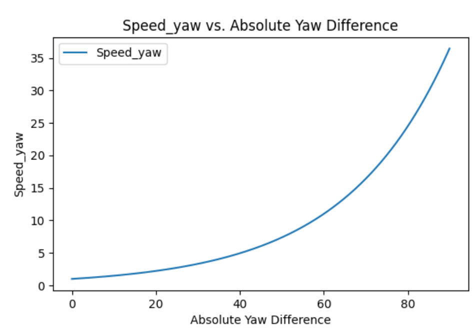


Fig. 3. Speed

D. Eye blink detection and click event

Eye blink detection utilizes a Convolutional Neural Network (CNN) structure to identify the condition of eyes, differentiating between open and closed states. The model is trained using the MRL Eye Dataset, as illustrated in Figure 5, achieving a notable 96% accuracy in the unseen test dataset. The confusion matrix, depicted in Figure 6, provides further insights. The training procedure employs 80% of the dataset, with the remaining 20% allocated for validating the model's robustness.

The architecture comprises three convolutional layers, three max-pooling layers, one fully connected layer, a sigmoid output layer, and dropout layers strategically incorporated to mitigate overfitting risks Figure 5. Training spans 20 epochs with a batch size of 32, optimizing the model using the Adam optimizer and employing categorical cross-entropy as the loss function.

This meticulous training regimen ensures the model's proficiency in discerning eye blink states, making it a reliable tool for real-time applications where accurate eye state detection is paramount.

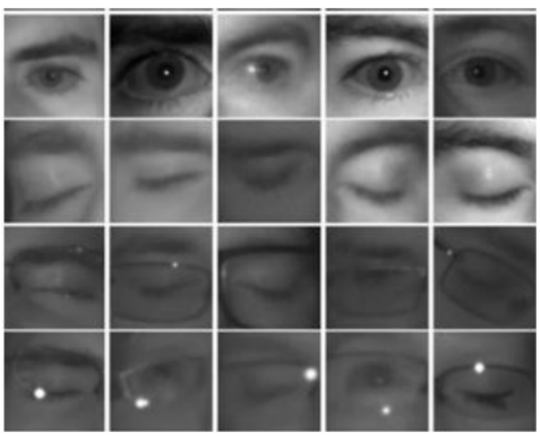


Fig. 4. MRL Eye Dataset

Algorithm 2 Eye Blink Detection

$pixel\_threshold, run\_threshold \leftarrow 900, 0.05$
if roi.size $>$ pixel_threshold **then** data_array.append(roi)
    if current_time $>$ run_threshold **then** $predictions$ = model.predict(np.array(data_array)) $mean$ = np.mean($predictions$)
        if $mean \leq 0.5$ **then**
            return "Eye blink detected"
        end if
    end if
end if

Prior to inputting a frame into the CNN model, the code identifies the region of interest (ROI) in the eye area using facial landmarks. This ROI is resized to 30x30 pixels and converted to grayscale. The CNN model then predicts the eye state based on this resized and grayscale ROI. The eye

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 28, 28, 32)        320

 max_pooling2d_6 (MaxPoolin  (None, 14, 14, 32)        0
 g2D)

 dropout_8 (Dropout)         (None, 14, 14, 32)        0

 conv2d_7 (Conv2D)           (None, 12, 12, 64)        18496

 max_pooling2d_7 (MaxPoolin  (None, 6, 6, 64)          0
 g2D)

 dropout_9 (Dropout)         (None, 6, 6, 64)          0

 conv2d_8 (Conv2D)           (None, 4, 4, 128)         73856

 max_pooling2d_8 (MaxPoolin  (None, 2, 2, 128)         0
 g2D)

 dropout_10 (Dropout)        (None, 2, 2, 128)         0

 flatten_2 (Flatten)         (None, 512)               0

 dense_4 (Dense)             (None, 128)               65664

 dropout_11 (Dropout)        (None, 128)               0

 dense_5 (Dense)             (None, 1)                 129

=================================================================
Total params: 158465 (619.00 KB)
Trainable params: 158465 (619.00 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig. 5. CNN Model Summary

```
Accuracy: 95.50%
Classification Report:
              precision    recall  f1-score   support

           0       0.92      1.00      0.96      1566
           1       1.00      0.91      0.95      1657

    accuracy                           0.96      3223
   macro avg       0.96      0.96      0.95      3223
weighted avg       0.96      0.96      0.95      3223
```

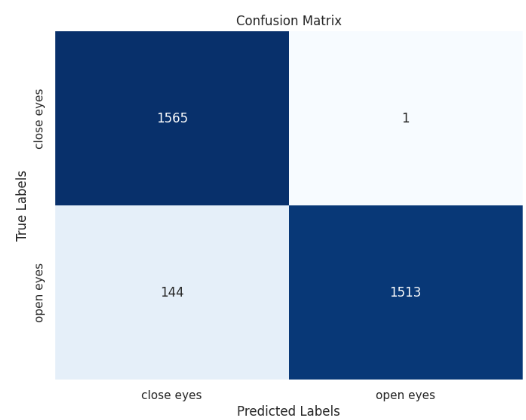


Fig. 6. Confusion Matrix

blink detection algorithm is outlined in Algorithm 2. Finally, a mouse click is triggered when 3 eye blinks are detected within a 2-second timeframe.

IV. Results

Prior to delving into the experiment results, let's examine the experiment setup outlined in Table I. The majority of the tasks were conducted on a laptop equipped with an Intel Core i7-7700HQ CPU, NVIDIA GEFORCE GTX 950M graphics card, and a webcam. The Convolutional Neural Network (CNN) model for eye detection was trained using the Kaggle platform.

TABLE I
EXPERIMENTAL SETUP

Item	Name/Value
OS	Windows 10
Processor	i7-7700HQ CPU
GPU	GTX 950M laptop
Speed	Approx. 8 FPS
IDE	Visual studio code and Kaggle
Language	Python
Device	Laptop
Camera	Webcam 640×480 (0.307MP)
OpenCV	4.9.0.80
Tensorflow	2.10.0
Dataset	MRL eye dataset

Testing occurs under three specific lighting conditions: daylight, lamplight, and darkness. Daylight testing takes place indoors on a sunny day with open window curtains. Lamplight testing is conducted indoors in a completely dark room using a lamp. Dark testing is performed indoors with no ambient light, relying solely on the screen light.

In addition, there is an additional test where the mouth is partially positioned outside the camera frame, capturing only the nose and upward. Users are positioned 50cm away from the webcam. We also conduct tests at large angles ranging from 60 to 90 degrees for direction recognition under daylight.

The objective of evaluating the approach under these varied lighting conditions and with partial facial visibility is to ensure its robustness across diverse environments. Three specific tests are employed: direction recognition, eye blink detection, and a YouTube video selection test.

The direction recognition test evaluates the system's capability to identify the direction of the head pose, and it expects the mouse cursor to move in correspondence with the recognized direction. During this test, users will be prompted to perform movements in the up, down, left, right, and center directions. The eye blink detection test assesses the system's effectiveness in detecting eye blinks.

In the YouTube video test, users are instructed to open YouTube from the bookmark saved on the browser's main page. They are then prompted to select a video, make it full screen, minimize the screen, and adjust the volume up and down.

The results of these tests are presented in Table II, III and IV.

TABLE II
LARGE ANGLES 60 TO 90 DEGREES

Lighting	Direction
Daylight	49/50, 96%

TABLE III
FULL FACE

Lighting	Direction	Eye blink	YouTube test
Daylight	50/50, 100%	19/20, 85%	70s
Lamplight	47/50, 94%	23/20, 85%	83s
Dark	46/50, 92%	17/20, 85%	178s

TABLE IV
MOUTH PARTIALLY OUTSIDE FRAME

Lighting	Direction	Eye blink	YouTube test
Daylight	49/50, 98	24/20, 80%	65s
Lamplight	41/50, 82%	18/20, 90%	144s
Dark	45/50, 90%	15/20, 75%	110s

The experimental findings suggest that the proposed method is effective, as all users were able to complete YouTube video test. The results demonstrate accurate direction estimation. While eye blink detection can initiate clicks, there is room for improvement in accuracy. In low-light environments, where the screen background is the only light source, the proposed approach operates only in white or light-colored videos due to the camera's inability to function in darkness. Additionally, when a laptop incorporates a second screen, the proposed approach is not capable, as the direction is determined when the user looks at the second screen, necessitating recalibration. Test results indicate that the proposed method functions well in the broader regions of the face, even those outside or hidden from the frame. In comparison to previous approach [17], the proposed method enhances mouse control capabilities at larger angles.

V. CONCLUSION

In summary, our paper introduces a method that provides an alternative solution in Human-Computer Interaction (HCI). The proposed approach integrates 3DDFA, head pose estimation, and CNN-based eye blink detection to propose a robust and inclusive method for mouse control. To refine the approach, we aim to improve the accuracy of eye blink detection, particularly in scenarios where users wear glasses. This enhancement involves collecting images of users with glasses and training the CNN with this updated dataset. Furthermore, incorporating head pose calibration into the system can contribute to its improvement, making it more adaptable for a broader spectrum of users.

REFERENCES

[1] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3d solution," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2016, pp. 146–155. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.23

[2] A. HeydariGorji, S. M. Safavi, L. Cheng-Ting, and P. H. Chou, "Head-mouse: A simple cursor controller based on optical measurement of head tilt," Jun 24 2020, copyright - © 2020. This work is published under http://arxiv.org/licenses/nonexclusive-distrib/1.0/ (the "License"). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2020-06-26. [Online]. Available: https://ezproxy.lincoln.ac.nz/login?url=https://www.proquest.com/working-papers/head-mouse-simple-cursor-controller-based-on/docview/2417176599/se-2

[3] "A human–computer interface replacing mouse and keyboard for individuals with limited upper limb mobility," *Multimodal Technologies and Interaction*, vol. 4, no. 4, p. 84, 2020, copyright - © 2020. This work is licensed under http://creativecommons.org/licenses/by/3.0/ (the "License"). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2023-11-30. [Online]. Available: https://ezproxy.lincoln.ac.nz/login?url=https://www.proquest.com/scholarly-journals/human-computer-interface-replacing-mouse-keyboard/docview/2466148197/se-2

[4] H. A. Alhamzawi, "Control mouse cursor by head movement: Development and implementation," *Applied Medical Informatics*, vol. 40, no. 3, pp. 39–44, 2018, copyright - © 2018. This work is published under NOCC (the "License"). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2023-11-19; SubjectsTermNotLitGenreText - United States–US. [Online]. Available: https://ezproxy.lincoln.ac.nz/login?url=https://www.proquest.com/scholarly-journals/control-mouse-cursor-head-movement-development/docview/2186572971/se-2

[5] M. Nabati and A. Behrad, "Camera mouse implementation using 3d head pose estimation by monocular video camera and 2d to 3d point and line correspondences," in *2010 5th International Symposium on Telecommunications*, 2010, pp. 825–830.

[6] B. Dhananjay, S. Rishabh, R. Narendra, M. Praphull, A. K. Uttam, and S. k. Arjaria, "Responsive human-computer interaction model based on recognition of facial landmarks using machine learning algorithms," *Multimedia Tools and Applications*, vol. 81, no. 13, pp. 18 011–18 031, 05 2022, copyright - © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022; Last updated - 2023-11-30. [Online]. Available: https://ezproxy.lincoln.ac.nz/login?url=https://www.proquest.com/scholarly-journals/responsive-human-computer-interaction-model-based/docview/2660203431/se-2

[7] M. Patacchiola and A. Cangelosi, "Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods," *Pattern Recognition*, vol. 71, pp. 132–143, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317302327

[8] J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li, "Towards fast, accurate and stable 3d dense face alignment," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[9] J. Guo, X. Zhu, and Z. Lei, "3ddfa," https://github.com/cleardusk/3DDFA, 2018.

[10] A. Królak and P. Strumio, "Eye-blink detection system for human-computer interaction," *Universal Access in the Information Society*, vol. 11, no. 4, pp. 409–419, 11 2012, copyright - Springer-Verlag Berlin Heidelberg 2012; Document feature - ; Last updated - 2023-12-05. [Online]. Available: https://ezproxy.lincoln.ac.nz/login?url=https://www.proquest.com/scholarly-journals/eye-blink-detection-system-human-computer/docview/1115061571/se-2

[11] R. Gawande and S. Badotra, "Deep-learning approach for efficient eye-blink detection with hybrid optimization concept," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, 2022, copyright - © 2022. This work is licensed under https://creativecommons.org/licenses/by/4.0/ (the "License"). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2023-11-25. [Online]. Available: https://ezproxy.lincoln.ac.nz/login?url=https://www.proquest.com/scholarly-journals/deep-learning-approach-efficient-eye-blink/docview/2695608282/se-2

[12] C. Dewi, R.-C. Chen, X. Jiang, and H. Yu, "Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks," *PeerJ Computer Science*, Apr 18 2022, copyright - © 2022 Dewi et al. This is an open access article distributed under the terms of the Creative Commons Attribution License: https://creativecommons.org/licenses/by/4.0/ (the "License"), which permits unrestricted use, distribution, reproduction and adaptation in any medium and for any purpose provided that it is properly attributed. For attribution, the original author(s), title, publication source (PeerJ Computer Science) and either DOI or URL of the article must be cited. Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2023-12-03. [Online]. Available: https://ezproxy.lincoln.ac.nz/login?url=https://www.proquest.com/scholarly-journals/adjusting-eye-aspect-ratio-strong-blink-detection/docview/2651899807/se-2

[13] B. Dhananjay, S. Rishabh, R. Narendra, M. Praphull, A. K. Uttam, and S. k. Arjaria, "Responsive human-computer interaction model based on recognition of facial landmarks using machine learning algorithms," *Multimedia Tools and Applications*, vol. 81, no. 13, pp. 18 011–18 031, 05 2022, copyright - © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022; Last updated - 2023-11-30. [Online]. Available: https://ezproxy.lincoln.ac.nz/login?url=https://www.proquest.com/scholarly-journals/responsive-human-computer-interaction-model-based/docview/2660203431/se-2

[14] M. Patacchiola and A. Cangelosi, "Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods," *Pattern Recognition*, vol. 71, pp. 132–143, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317302327

[15] J. Huang, Z. Zhang, G. Xie, and H. He, "Real-time precise human-computer interaction system based on gaze estimation and tracking," *Wireless Communications and Mobile Computing (Online)*, vol. 2021, 2021, copyright - Copyright © 2021 Junhao Huang et al. This work is licensed under http://creativecommons.org/licenses/by/4.0/ (the "License"). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2023-12-05. [Online]. Available: https://ezproxy.lincoln.ac.nz/login?url=https://www.proquest.com/scholarly-journals/real-time-precise-human-computer-interaction/docview/2600073492/se-2

[16] I. Jahan, K. Aslam Uddin, A. M. Saydul, U. M. M Saef, Z. K. Tanvir, M. Masud, S. Aljahdali, and A. K. Bairagi, "4d: A real-time driver drowsiness detector using deep learning," *Electronics*, vol. 12, no. 1, p. 235, 2023, copyright - © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2023-11-23. [Online]. Available: https://ezproxy.lincoln.ac.nz/login?url=https://www.proquest.com/scholarly-journals/4d-real-time-driver-drowsiness-detector-using/docview/2761105044/se-2

[17] D. Li and W. Pedrycz, "A central profile-based 3d face pose estimation," *Pattern Recognition*, vol. 47, no. 2, pp. 525–534, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320313003178