SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I3P114

Volume 11 Issue 3, 130-140, March 2024

Original Article

Real-Time Head Pose Estimation Using Haar Cascade Classifier for Visual Attention Application

Muhammad Syaqir Aiman Rosli[1], Rusnani Yahya[1], Rozita Jailani[1,2*], Nur Khalidah Zakaria[1], Heru Supriyono[3]

[1]*School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA (UiTM), Malaysia.*
[2]*Integrative Pharmacogenomics Institutes (iPROMISE), Universiti Teknologi MARA, Puncak Alam, Malaysia.*
[3]*Department of Electrical Engineering, Universitas Muhammadiyah Surakarta, Indonesia.*

**Corresponding Author : rozitaj@uitm.edu.my*

Received: 11 January 2024 Revised: 17 February 2024 Accepted: 15 March 2024 Published: 31 March 2024

Abstract *- This research paper explores the development and evaluation of a real-time head pose estimation algorithm for visual attention application, addressing challenges in varying conditions, complex poses, and partial occlusion. Leveraging the OpenCV library and Haar cascade classifier, the algorithm was implemented and tested with a smartphone camera setup. The study involved 10 subjects under different conditions, revealing high accuracy in controlled scenarios. The methodology incorporated innovative features, including cascade classifiers for diverse facial orientations. Results indicated varying accuracy influenced by environmental factors and subject movements. The average accuracy of the developed algorithm applied to various testing conditions is more than 84% for head pose estimation and more than 90% for visual attention. The findings contribute insights into algorithm efficacy, showcasing potential applications in fields of healthcare, therapy, driving monitoring and others. Overall, this research lays a foundation for robust head pose estimation systems with real-world adaptability.*

Keywords *- Head pose estimation, OpenCV, Haar cascade classifier, Python, AdaBoost.*

1. Introduction

Advancements in Artificial Intelligence (AI) and computer vision have yielded substantial breakthroughs across diverse fields. Notably, head pose estimation has emerged as a focal point, playing a crucial role in numerous applications, including healthcare monitoring, online meetings, teaching, attendance management systems, and driver monitoring [1, 2]. Despite its significance, existing head pose estimation algorithms encounter challenges in achieving real-time detection in video or live camera feeds, especially in scenarios characterized by variable conditions, complex poses, and partial occlusion [3, 4]. The Haar cascade classifier, renowned for its efficacy in object detection, is specifically designed for identifying faces or objects in both images and real-time videos. Widely acknowledged as a proficient classifier for head pose estimation applications [5], it excels in achieving a high detection rate for human frontal faces while enhancing processing speed [6, 7].

Additionally, it proves valuable as an image preprocessing technique to improve the quality of input images. Face alignment is then performed to standardize facial orientation across different images [8]. The algorithm adeptly determines whether the subject is facing straight, turning right, or turning left toward the camera. To ensure real-time applicability, the implementation incorporates face tracking, allowing the algorithm to accurately follow and track detected faces as they move or turn within a video or image sequence [9]. All real-time processes and observed data during subject observation are meticulously logged to facilitate evaluation. The final stage involves comprehensive data analysis and rigorous algorithm validation to assess performance under various challenging conditions.

Acknowledging these limitations, this research aims to develop a head pose estimation algorithm capable of efficiently handling real-time video or live camera feeds with varying conditions, poses, and occlusion [10].

The algorithm's effectiveness will be validated through precise testing, with each subject undergoing three different conditions. To achieve these goals, a systematic approach to head pose estimation, utilizing the Open-Source Computer Vision (OpenCV) library in Python, will be adopted [11]. The process includes calibration for accurate positioning of the subject's face and head, followed by video recording using a webcam and storing it on a computer or laptop.

2. Methodology

This section will provide a detailed explanation of the methodology employed in developing the head pose estimation algorithm system.

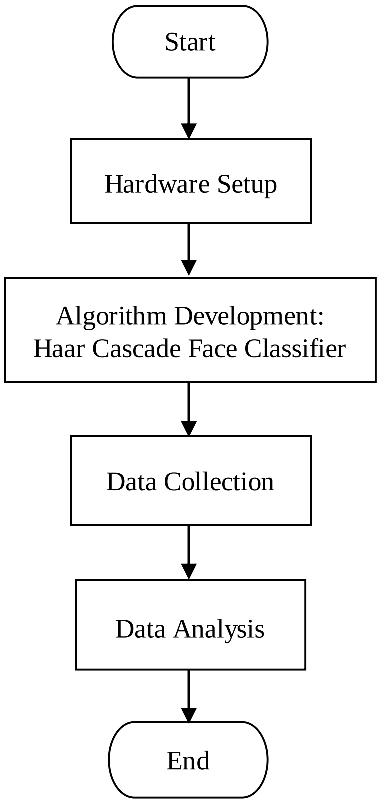

Fig. 1 General flowchart of the system

Figure 1 serves as the comprehensive flowchart for the entire study, outlining the sequence of activities from inception to conclusion. The study commences with the hardware setup, employing a smartphone camera connected to coding. The phone is securely mounted on a tripod, strategically positioned at an appropriate height and distance to accommodate the subject's position.

OpenCV, a prominent open-source library for computer vision and image processing, assumes a pivotal role in implementing the head pose estimation algorithm in conjunction with the Haar cascade classifier for this project [12].

In the data collection phase, 10 subjects, all students from Universiti Teknologi MARA (UiTM), Shah Alam, aged between 20 and 24 years old, participated in the study. Each subject underwent two categories of data collection. Firstly, the subjects executed movements as instructed by the researcher. Additionally, subjects engaged in random movements. Both categories were conducted under three different conditions: face without obstacles, with spectacles, and with a cap or hat. As a result, a total of 30 data points were collected.

Once the recorded video and data were successfully saved, the subsequent phase involved data analysis. The process aimed to evaluate the system's performance by calculating the percentage of accuracy. The predicted data generated by the system were compared and validated against the recorded video. True detections, where the system's predictions aligned with the actual movements, contributed to the accuracy percentage. Conversely, instances of false detections were considered errors, providing insights into the system's limitations and areas for improvement.

2.1. Hardware Setup

The initial phase of the study focuses on the essential hardware setup, critical to the successful implementation of the head pose estimation system. This setup integrates a smartphone camera with the system framework. Leveraging the OpenCV library, which simplifies access to camera feeds through dedicated functions, the study designates the default camera as a smartphone camera, boasting a superior resolution of 108 megapixels for capturing real-time video frames.

This choice is preferred over the laptop's built-in webcam due to its comparatively lower resolution. To ensure stability and optimal positioning, the smartphone is securely mounted on a tripod and strategically placed at an appropriate height and distance, considering the subject's position, to achieve effective head pose estimation.

To facilitate the subject's involvement, a chair is positioned in front of the tripod and camera. Subjects are required to sit comfortably in this chair before commencing study. Maintaining a distance of 1 meter or less between the chair and the camera is essential for ensuring the accurate face detection capabilities of the system. This distance parameter is carefully established to optimize the system's performance. Throughout the study, the researcher monitors progress on a laptop connected to the smartphone camera via the same Wi-Fi connection.

Moreover, emphasizing the importance of good lighting conditions, the study acknowledges that sufficient illumination significantly contributes to the proper functionality of the head pose estimation system. Overall, this precise hardware setup forms the foundation for conducting reliable and effective head pose estimation studies.

2.2. Haar Cascade Classifier

The Haar cascade classifier is a machine learning-based object detection method that employs Haar-like rectangular features and the integral image method to expedite feature evaluation during bottom feature extraction. This classifier trains a robust model through the AdaBoost algorithm and subsequently cascades. The typical process of utilizing the Haar cascade classifier for head pose estimation entails sliding a sub-window continuously over the image window to be detected, computing the feature of the region at each position, and then applying the cascade classifier to filter the feature. If the feature successfully passes all filtering stages, the region is identified as a face [11].

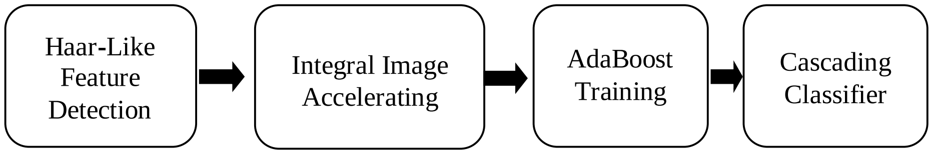

Fig. 2 Four stages of the Haar cascade classifier

The Haar cascade classifier algorithm is trained on images through a cascade function comprising four main stages. First, it collects Haar-like features through detection. Second, it expedites feature evaluation using integral images. Third, it employs the AdaBoost algorithm to train a robust classifier capable of distinguishing between faces and non-faces. Finally, it cascades all strong classifiers together. Figure 2 visually represents this process.

The Haar cascade classifier algorithm involves four main stages. In the first stage, Haar-like features are collected, which are categorized into three types: edge features (A and B), linear features (C), and four-rectangle features (D), as illustrated in Figure 3. These features are amalgamated into a feature template comprising two rectangles of white and black. The feature value of the template is defined as the sum of white rectangle pixels minus the sum of black rectangle pixels [14, 15]. When calculating the feature value from both the face area and the non-face area, a greater difference signifies a more effective feature template in distinguishing faces.

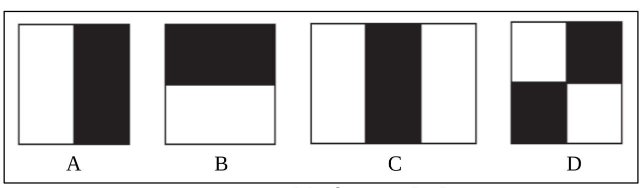

Fig. 3 Haar-like features [13]

In the Haar cascade classifier algorithm, the rectangular feature in the image sub-window is obtained by translating and expanding the feature template. The position and size of rectangular features in the image can be altered arbitrarily, making the rectangular feature value only dependent on the category, position, and size of the rectangular template [16].

Consequently, a small detection window generates a substantial number of rectangular features due to the variations in these three factors. For instance, the number of rectangular features in a detection window with a size of 24x24 pixels can reach 160,000 [16].

$$ii(i,j) = \sum_{k\leq i\; l\leq j} f(k,l) \quad (1)$$

The second step of the Haar cascade classifier algorithm involves utilizing the integral image to expedite feature evaluation in the face of a large number of rectangular features [17]. To enhance the evaluation speed and prevent redundant calculations in repeated regions, it is essential to construct an integral image. The integral image is constructed by calculating the sum of regional pixels at once using a dynamic programming algorithm based on the original image, treating pixels as units.

By indexing the integral image, the need for repeated calculations is eliminated, significantly accelerating the calculation speed. The integral image serves as a matrix description method for regional global information, and its construction method is defined by the formula (1). The formula denotes that the value ii(i, j) at the position (i, j) is the sum of all pixels f (k, l) in the upper-left corner of the original image (i, j) [16].

The third step of the Haar cascade classifier algorithm involves building a strong classifier to distinguish between faces and non-faces, implemented through the AdaBoost algorithm [11, 18]. The AdaBoost algorithm constructs a robust classifier by combining several weak classifiers, representing an enhancement of the boosting algorithm based on the theoretical foundation of the PAC learning model [11].

Weak classification results are generated by moving a window over the input image and computing Haar features for each segment [18]. In the process of obtaining multiple weak classifiers through iterative training rounds, the initial weights of each training sample are set, and the weights are iteratively updated based on the classification results of each training. These updated weights are then used as the initial weights for the next round of training.

A new weak classifier is added in each training round until a predetermined low enough error rate is reached or a specified maximum number of iterations is achieved [11]. The rationale behind updating sample weights is to increase the weights of misclassified samples in each training, focusing on learning samples with poor classification effects in later rounds to improve their classification ability.

$$\alpha_m = \frac{1}{2} log \frac{1-e_m}{e_m} \quad (2)$$

In the Haar cascade classifier algorithm, each round of weak classifier training aims to learn better hyperparameters, resulting in a smaller classification error for all training samples in the classifier. This iterative process leads to a progressively lower overall error rate for the model. The weight of a weak classifier, denoted as m, is determined by the classification error rate me of the classifier Gm(x), as shown in formula 2. When combining the weak classifiers obtained from each training round into a strong classifier in the Haar cascade classifier algorithm, the weak classifier with a lower classification error rate carries a greater weight in the final strong classifier. It exerts a more significant influence on the decision-making process. Conversely, weak classifiers with higher error rates contribute less. The formula for multiple weak classifiers forming a strong classifier is presented in formula 3.

$$G_{(x)} = sign(\sum_{m=1}^{M} \alpha_m G_m(x)) \quad (3)$$

In the Haar cascade classifier algorithm, formula 3 signifies that M weak classifiers are weighted and voted based on the weight α_m of each weak classifier Gm(x) to constitute a strong classifier G(x) [11]. Consequently, multiple weak classifiers, each concentrating on different characteristics of various samples, can collectively create a strong classifier with enhanced classification performance through weighted combinations according to the classification error rates.

The fourth step of the Haar cascade classifier algorithm involves using a screening cascade to cascade strong classifiers together, enhancing accuracy. The cascaded classifier, depicted in Figure 3, comprises multiple strong classifiers, each encompassing several weak classifiers. Each weak classifier is obtained through the AdaBoost algorithm combined with Haar-like feature training.

As the detection window passes through all strong classifiers, it is deemed a face; otherwise, it is considered a non-face. Both the strong classifier and each weak classifier follow tree structures. Due to the high accuracy of each strong classifier in distinguishing non-faces, once it is determined that the target in the detection window is not a face, the subsequent strong classifiers are not invoked, significantly reducing detection time.

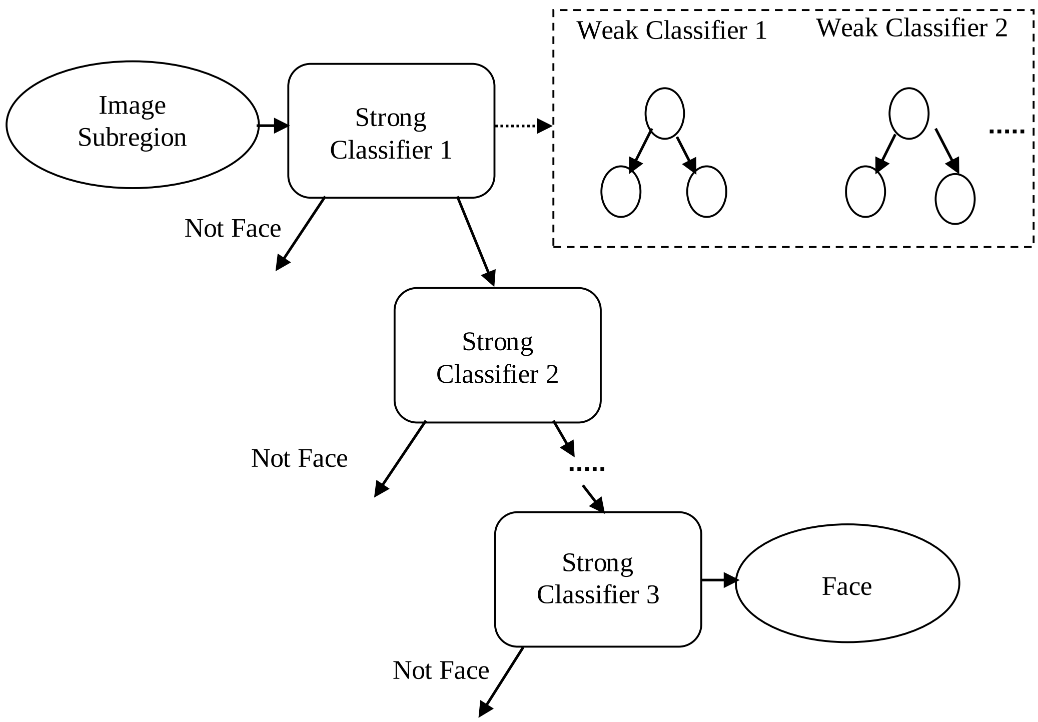

Fig. 4 Haar cascade classifier

2.3. Data Collection

During the data collection phase, a cohort of 10 UiTM students aged between 20 and 24 years participated in the study. Each subject underwent data collection under three conditions: a face without obstacles, wearing spectacles, and wearing a hat or cap.

For each condition, the subject was categorized into two groups: predicted and random. Each category required 2 minutes to complete. In the predicted category, the subject was instructed to sit comfortably on the provided chair. When the system program was initiated, the researcher directed the subject to perform face movements. This first category was further divided into four parts, each lasting 30 seconds.

The subject was instructed to look forward, turn right, look forward again, and then turn left. It was specified that each turn of the face must exceed 70 degrees. For the second category, subjects were allowed to move their faces randomly in front of the camera for a duration of 2 minutes.

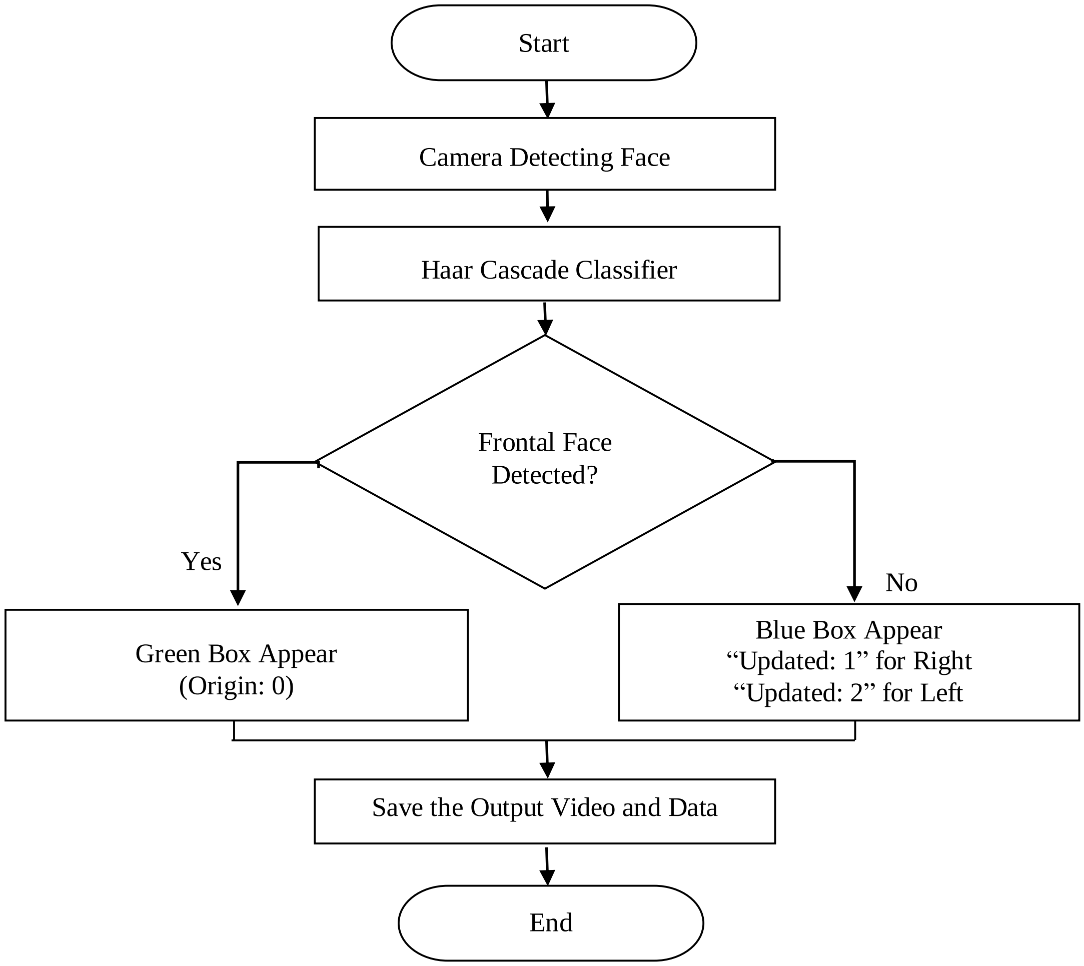

Fig. 5 Flowchart of data collection

According to Figure 5, upon running the system program, the camera initiates head orientation detection of the subject. After capturing the face, it is loaded into the Haar cascade classifier provided by OpenCV. The study specifically employs three classifiers designed to identify faces in frontal, right, and left orientations. Rectangles are drawn on the original frame for each detected face, distinguishing between frontal faces (green box) and right/left faces (blue box).

The recording takes place at specified intervals, enhancing the efficiency of data collection. To visualize the head pose estimation process, the study employs a system function to generate an output video file in '.mp4' format. For further analysis, the collected data, enriched with timestamps and corresponding updated values of face orientation, is exported to an Excel file and saved on the hard drive.

2.4. Data Analysis

The study exclusively assesses the system's performance by calculating the percentage of accuracy. Predicted data generated by the system may not achieve full accuracy, especially when the head turn does not exceed 70 degrees. However, the system categorizes it as a Right or Left head turn. Therefore, the predicted data is compared and validated against the recorded video, as actual data can only be evaluated from recorded videos. True detections, where the system's predictions align with the actual movements, contribute to the accuracy of the system. Conversely, instances of false detections are considered errors, providing insights into the system's limitations and areas for improvement.

Once the data on true and false detections has been summarized, the number of true detections will be divided by the total amount of data for each condition for a subject to obtain a percentage of accuracy. From all the obtained percentages, accuracy graphs for each condition will be produced. An overall accuracy percentage graph will also be generated to identify which condition has the highest accuracy percentage.

3. Results and Discussion

This section will discuss the results obtained from this study in detail. The accuracy of the algorithm with a different set of experiments will be analysed.

3.1. Hardware Setup

Fig. 6 Hardware setup

Figure 6 illustrates the experimental setup, wherein a smartphone camera is utilized for recording due to its high resolution. The integration involves connecting the smartphone camera to the system framework, ensuring the recorded video is of high quality and facilitating improved head pose estimation. Throughout the study, a laptop must be connected to the smartphone camera through the same Wi-Fi connection for progress monitoring. The built-in webcam on the laptop is not considered in this case, as its resolution is comparatively lower.

The phone is securely mounted on a tripod and strategically positioned at an appropriate height and distance for the subject's position. The distance parameter between the camera and the subject must not exceed 1 meter. Additionally, the study underscores the significance of good lighting conditions, recognizing that ample illumination significantly enhances the system's performance.

3.2. Haar Cascade Classifier

The standard Haar cascade classifier is traditionally trained to recognize faces oriented straight towards the camera, focusing on features indicative of a frontal view, as outlined in the methodology. In this study, the improved algorithm extends the functionality of the Haar cascade classifier to include face orientation detection by integrating additional XML files for right and left profiles.

This enhancement augments the original frontal face detection capability, enabling the system to identify faces turning right or left. While frontal face detection remains essential for establishing the baseline, the added right and left face cascades to facilitate a more comprehensive analysis of facial orientations. Operating on the principles outlined in the methodology, the Haar cascade classifier utilizes Haar-like rectangular features and integral images to evaluate features during the bottom feature extraction process efficiently. Its strength lies in employing the AdaBoost algorithm, which combines multiple weak classifiers to form a robust model capable of distinguishing between faces and non-faces. Iterative training rounds progressively reduce overall error rates, resulting in a strong classifier with improved classification performance.

The cascade structure, as depicted in Figure 4, remains a crucial aspect of the enhanced algorithm. The screening cascade incorporates multiple strong classifiers, each comprising several weak classifiers. Decision-making involves a weighted combination of weak classifiers based on their classification error rates. This cascade structure, organized in a tree-like manner, accelerates the detection process by effectively filtering out non-face regions.

Furthermore, the algorithm seamlessly integrates face orientation detection. With the inclusion of right and left face cascades, the system can now discern faces turning in either direction. This extension aligns with the foundational principles of the Haar cascade classifier, adapting to variations in face orientation through a multitude of rectangular features within the detection window. The improved algorithm maintains the efficiency and accuracy of the original classifier while expanding its applicability to real-time scenarios where face orientations are crucial.

In conclusion, the enhanced Haar cascade classifier not only preserves its core features but also extends its capabilities to detect faces in diverse orientations. Integration of additional XML files for right and left face profiles enhances the algorithm's versatility, making it invaluable for applications requiring real-time head pose estimation and visual attention analysis.

3.3. Data Collection

Statistical information has been collected through consent forms completed by 10 participants in this study. The group consists of 7 male students and 3 female students, all aged between 20 and 24. This age range is selected to ensure stability in the average percentage of accurate data collected. Participants' height falls within the range of 160 to 175 cm, while their body weight ranges from 55 to 85 kg. It is noteworthy that weight is a crucial factor, as overweight subjects may introduce errors in the system during the detection process.

(a) **(b)**

(c)

Fig. 7 Head pose estimation with different orientations

Fig. 8 Improper head turn (<70 degrees)

Figure 7(a) depicts the appearance of a green box labeled "Origin: 0" when the system detects a frontal face. In instances where the system detects face orientation turning right or left, Figure 7(b) displays a blue box labeled "Updated: 1" for right-side detection, and Figure 7(c) shows "Updated: 2" for left-side detection. If the subject executes a fast movement or the turn does not exceed 70 degrees, both green and blue boxes will simultaneously appear, exemplified in Figure 8.

In this scenario, the system captures the most stable detection of face orientation. Specific naming conventions are employed for the output video and data file for each subject and condition, such as "Face_01" for subject 1. The system post-recording contains timestamps for every 2 seconds and corresponding updated values of face orientation.

3.4. Data Analysis

This section will provide a detailed explanation of the data derived from the study. The analysed data focuses on the system accuracy percentage, categorized into two distinct groups: controlled and random; to validate the algorithm. Then, the analysis of the effectiveness of the algorithm in detecting visual attention.

3.4.1. Controlled Head Pose

This section will analyze the results obtained from the controlled setting, where subjects followed the instructions provided by the researcher. The experiment was conducted three times, each with different conditions (face without obstacles, with spectacles, and with a cap or hat).

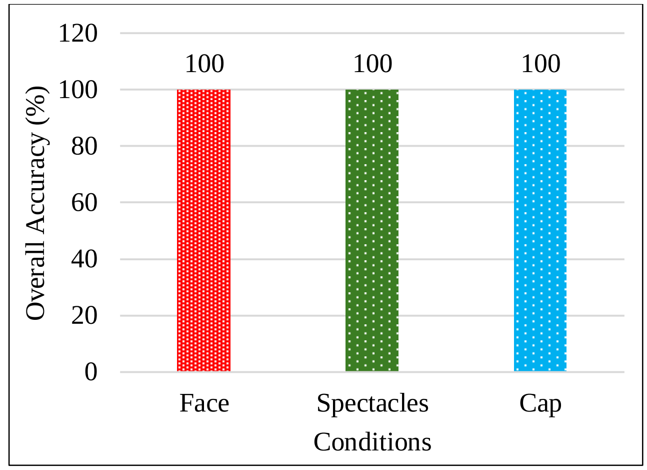

Fig. 9 Overall accuracy (%) for the controlled category with 3 different conditions

Figure 9 above illustrates the overall percentage for each condition (face without obstacles, with spectacles, and with a cap or hat) conducted in the study involving 10 subjects. In this experiment setting, the subjects were instructed to slowly turn their heads to the right or left until a full head turn (around 70 degrees) was achieved. The results obtained from this setting, as depicted in Figure 9, show that all head poses were estimated correctly with 100% accuracy for all three conditions. This indicates that the system successfully detected and identified face orientations and head poses according to the given instructions with complete precision.

3.4.2. Random Head Pose

This section will analyze the results obtained from the random setting, where subjects had the freedom to move their heads freely and randomly without following any specific instructions. The experiment was conducted three times, each with different conditions (face without obstacles, with spectacles, and with a cap or hat).

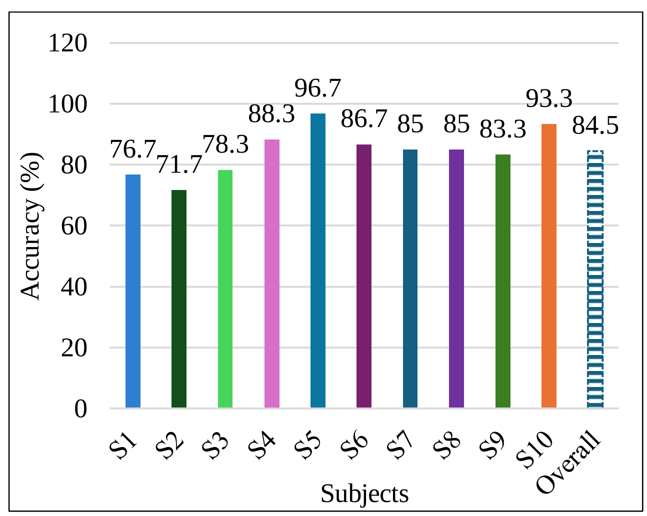

Fig. 10 Accuracy (%) for a random category with a face without obstacles

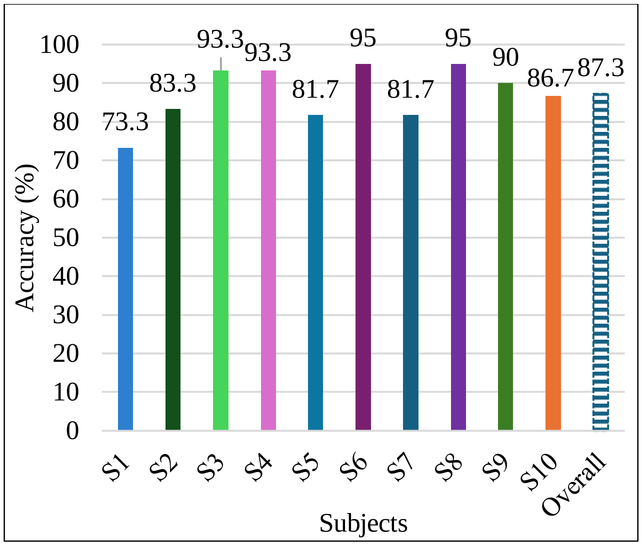

Fig. 11 Accuracy (%) for random category while subjects wearing spectacle

Figure 10 presents accuracy percentages in category 2, where subjects are allowed random head movements during recording and face with no obstacles. Accuracy varies among subjects in this category due to factors influencing the system's ability to capture and identify random head movements. Subjects 1 to 7 are male, and subjects 8 to 10 are female.

Examining Figure 10, Subject 5 demonstrates the highest accuracy, while Subject 2 exhibits the lowest. Validation indicates that environmental disturbances contribute to this difference. Subject 2 had items behind them during recording, unlike Subject 5, whose background was more spacious and clearer. Environmental disturbances impact system accuracy differently for each subject. The average accuracy for all 10 subjects while executing random head poses is 84.5%, indicating that the algorithm effectively captures head pose positions.

Figure 11 illustrates the accuracy for the random category with all subjects wearing spectacles during data collection. Notably, despite gender differences between Subject 6 (male) and Subject 8 (female), both achieved the same accuracy percentage, 95%. In this study, all female subjects wore a hijab, suggesting that the combination of hijab and glasses does not hinder the system's ability to achieve high true detection.

In another observation, subject 3 recorded varying accuracy percentages in different conditions, as evident in both Figures 10 and 11, with a percentage difference of 15%. Upon reviewing the recorded videos of subject 3, it was noted that the subject exhibited less movement while wearing spectacles compared to when not wearing anything on the face. Additionally, the subject made faster movements but did not complete a turn of the face beyond 70 degrees during random movement with no face obstacle. This incomplete movement affected the system's ability to capture the head pose correctly, resulting in lower accuracy in condition 1 compared to condition 2. However, the average accuracy for this category and condition is slightly higher than in condition 1, leading to the conclusion that wearing spectacles does not affect the detection of head pose.

Figure 12 illustrates the accuracy percentage when the system detects a face while subjects are wearing a hat. Analysis reveals that Subjects 5 and 7 achieved the highest accuracy, both reaching 95% for this condition. The similarity is attributed to the nearly perfect movements executed by these subjects when turning their faces to the front, right, and left, even in random sequences and while wearing a hat.

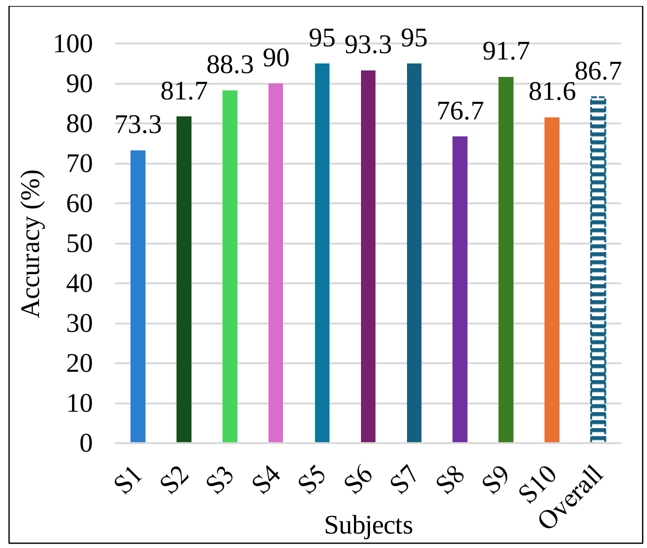

Fig. 12 Accuracy (%) for random category while subjects were wearing caps or hat

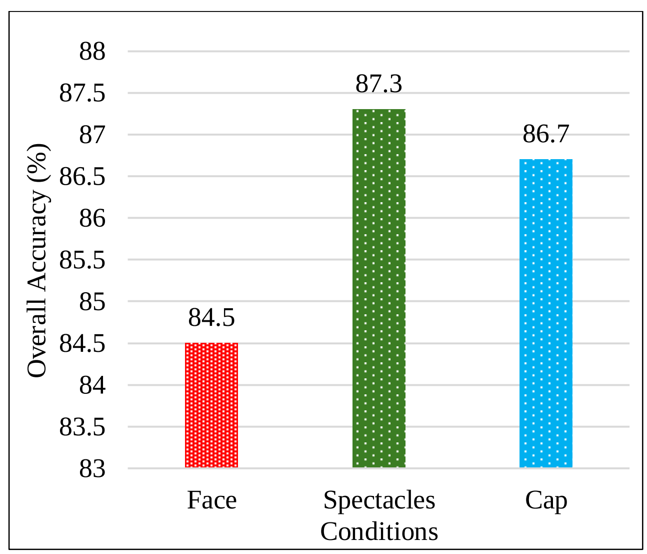

Fig. 13 Accuracy (%) for random category for all three categories (face without obstacles, spectacle, and cap/hat)

In contrast, Subject 1 presents a different scenario with numerous fast movements during the recording process. These rapid movements disrupt the system's ability to detect the right and left sides of the subject's face due to imperfect turning angles. Additionally, this subject wears a hat in an environment with poor lighting conditions, resulting in numerous false detections considered as errors in all three conditions.

Figure 13 displays the average accuracy across all three conditions in the random category. Various factors have contributed to fluctuations in accuracy percentages for each subject. Condition 2 attained the highest overall percentage compared to the others. This is attributed to individual validations on all recorded videos, revealing that most subjects performed movements more effectively in condition 2 than in condition 1. In the final condition, the overall percentage slightly decreases from condition 2 due to subjects wearing a hat, which can obstruct light towards the face during the recording process. However, this factor has minimal impact and does not impede the system from detecting head turns when the head position is optimal. Overall, the accuracy for this experiment exceeds 84%, indicating that the proposed algorithm effectively estimates head pose accuracy, and obstacles such as spectacles, hats, and caps have minimal impact on the accuracy of detecting head position.

3.4.3. Visual Attention

In this section, utilizing the dataset from the random setting under three distinct conditions (face without obstacles, with spectacles, and with a cap or hat), the algorithm's efficacy in detecting visual attention is analysed. The accuracy of the algorithm is assessed by calculating prediction outputs only when subjects are directly facing the camera (visual attention). The algorithm's results are validated against video recordings for accuracy assessment.

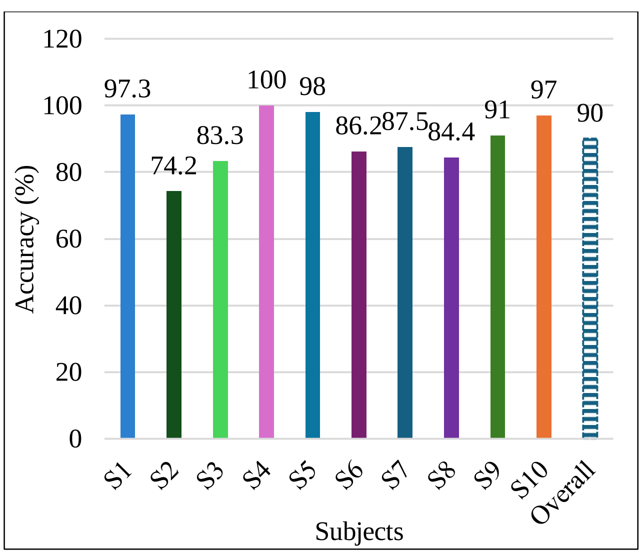

Fig. 14 Accuracy (%) for visual attention with a face without obstacles

Figure 14 illustrates the accuracy for all 10 subjects during the visual attention duration, where the algorithm automatically detects when the subject faces directly towards the camera. The graph indicates an average accuracy of 90% across the 10 subjects. Notably, Subject 2 exhibits a false detection rate of 25% as the subject briefly faces the camera during data collection, leading the algorithm to predict that visual attention is not consistently maintained throughout the collection period. Consequently, the definition of visual attention duration may be tailored based on the specific application requirements. Figure 15 presents the accuracy of visual attention detection obtained from subjects wearing spectacles during data collection. The graph indicates an average accuracy of 92% across all 10 subjects. This demonstrates that the system is capable of accurately detecting the visual attention period, even for subjects wearing spectacles.

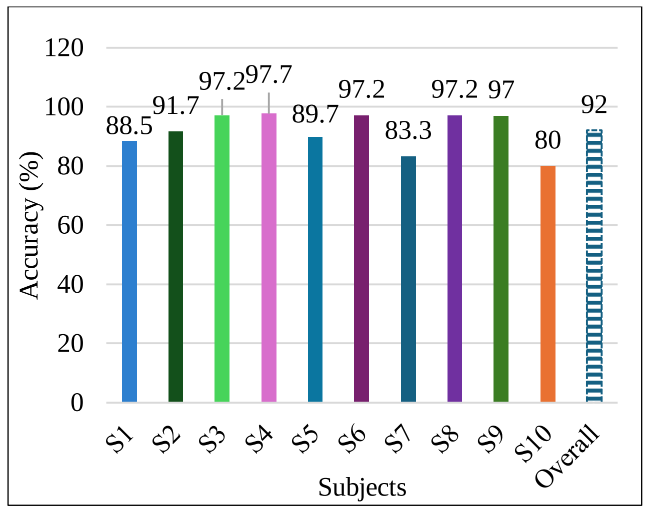

Fig. 15 Accuracy (%) for visual attention while subjects wearing spectacle

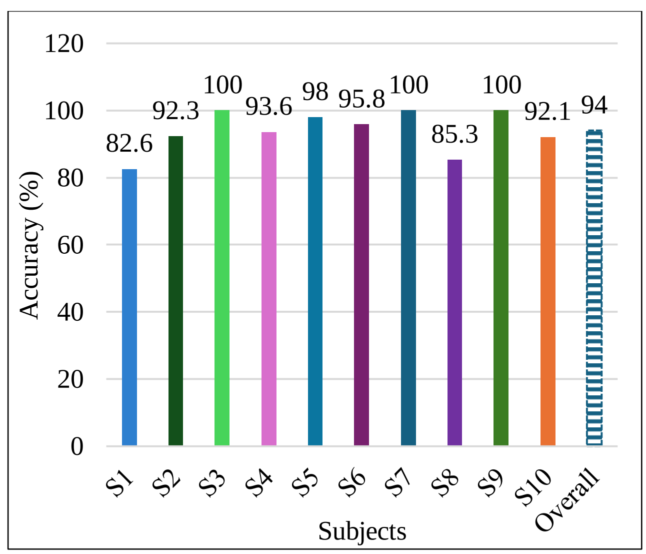

Fig. 16 Accuracy (%) for visual attention while subjects wearing hat/cap

Figure 16 illustrates the accuracy of visual attention detection obtained from subjects wearing hats or caps during data collection. The graph indicates an average accuracy of 94% across all 10 subjects. Notably, the results from this experiment surpass those of the previous two conditions (face without obstacles and spectacle). It is important to note that, during these three conditions, subjects freely move their heads and faces, with the rate of movement being uncontrolled and entirely random. The results affirm that the system is proficient in accurately detecting the visual attention period, even for subjects wearing hats or caps.

4. Conclusion

This study successfully estimated head pose positions in real time using the Haar cascade classifier. The enhanced Haar cascade classifier not only maintains its fundamental functionalities but also broadens its capabilities to accurately detect faces across a comprehensive range of orientations compared to the standard algorithm provided by OpenCV. The algorithm's performance was evaluated through two experiment settings: the first involved subjects following specific head movement instructions, while the second allowed subjects to move their heads freely. Both settings encompassed three conditions (face without obstacles, spectacle, and cap/hat) to assess the algorithm's effectiveness in the presence of facial obstructions. The findings indicate that the proposed algorithm achieves real-time head pose estimation with an accuracy exceeding 84% and successfully detects visual attention with an accuracy of 90% and above. It can be concluded that obstacles on the face minimally affect detection accuracy.

However, accuracy variations were influenced by factors such as environmental disturbances, subject movements, and accessory usage, suggesting potential areas for improvement. In summary, this study provides valuable insights into the efficacy of the developed head pose estimation algorithm, demonstrating its potential applicability across diverse scenarios, from controlled environments to real-world conditions.

Funding Statement

This research is funded by the British Council through the UK-Malaysia University Consortium Grant, project file: 100-TNCPI/INT 16/6/2 (072/2022) and (073/2022). Financial support also received from the Institute of Research Management Centre (RMC), Universiti Teknologi MARA (UiTM), project file: 100-RMC 5/3/SRP INT (047/2023) and (048/2023).

Acknowledgments

The authors wish to thank the Institute of Research Management Centre (RMC) and College of Engineering, Universiti Teknologi MARA (UiTM) Shah Alam, Malaysia, for the financial support, instrumentation and experimental facilities provided. The authors also wish to acknowledge the EUREKA Robotics Centre, Cardiff Metropolitan University, UK, for the research collaboration.

References

[1] Kruti Goyal, Kartikey Agarwal, and Rishi Kumar, "Face Detection and Tracking: Using OpenCV," *2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, pp. 474-478, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[2] Maliha Khan et al., "Face Detection and Recognition Using OpenCV," *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, pp. 116-119, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] M. Monica Dhana Ranjini et al., "Haar Cascade Classifier-Based Real-Time Face Recognition and Face Detection," *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, pp. 990-995, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Nazar Karpiuk, Halyna Klym, and Ivanna Vasylchychyn, "Facial Recognition System Based on the Haar Cascade Classifier Method," *2023 24th International Conference on Computational Problems of Electrical Engineering (CPEE)*, Grybów, Poland, pp. 1-4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Cheol-Ho Choi et al., "Face Detection Using Haar Cascade Classifiers Based on Vertical Component Calibration," *Human-Centric Computing and Information Sciences*, vol. 12, pp. 1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Ankit Srivastava et al., "A Survey of Face Detection Algorithms," *2017 International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, India, pp. 1-4, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[7] Bikang Peng, and Anilkumar Kothalil Gopalakrishnan, "A Face Detection Framework Based on Deep Cascaded Full Convolutional Neural Networks," *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, Singapore, pp. 47-51, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Li Cuimei et al., "Human Face Detection Algorithm via Haar Cascade Classifier Combined with Three Additional Classifiers," *2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, Yangzhou, China, pp. 483-487, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[9] Fachruddin et al., "Real Time Detection on Face Side Image with Ear Biometric Imaging Using Integral Image and Haar-Like Feature," *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Pangkal, Indonesia, pp. 165-170, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] An-Chao Tsai et al., "Occlusion Resistant Face Detection and Recognition System," *2020 8th International Conference on Orange Technology (ICOT)*, Daegu, Korea (South), pp. 1-4, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Xueyou Huang, and Xiang Cao, "Face Detection and Tracking Using Raspberry Pi Based on Haar Cascade Classifier," *2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Beijing, China, pp. 505-509, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] Xianghua Fan et al., "The System of Face Detection Based on OpenCV," *2012 24th Chinese Control and Decision Conference (CCDC)*, Taiyuan, pp. 648-651, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[13] Sharmeena Naido, and Rosalyn R. Porle, "Face Detection Using Colour and Haar Features for Indoor Surveillance," *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, Kota Kinabalu, Malaysia, pp. 1-5, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] L. Kh. Safiullina, A. Sh. Gabdullin, and I.V. Anikin, "Face Recognition in Biometric Systems Using Haar Cascade Classification," *2021 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, Omsk, Russian Federation, pp. 1-5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Asif Mohammed Arfi et al., "Real Time Human Face Detection and Recognition Based on Haar Features," *2020 IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, pp. 517-521, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Ratna Yustiawati et al., "Analyzing of Different Features Using Haar Cascade Classifier," *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Pangkal, Indonesia, pp. 129-134, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[17] Shahad Salh Ali, Jamila Harbi Al' Ameri, and Thekra Abbas, "Face Detection Using Haar Cascade Algorithm," *2022 Fifth College of Science International Conference of Recent Trends in Information Technology (CSCTIT)*, Baghdad, Iraq, pp. 198-201, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Wenxiang Yu et al., "A Depth Cascade Face Detection Algorithm Based on Adaboost," *2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, Beijing, China, pp. 103-107, 2016. [CrossRef] [Google Scholar] [Publisher Link]