GradFace: Attendance Registration System Using Face Recognition for The Graduate School, Kasetsart University

Sarach Tuomchomtam[1,2*], Suttipong Meuntaboot[2], Passapong Jittavanit[3] and Sornchai Laksanapeeti[2]

[1] Human-Centered AI Laboratory (KU-HCAI)
[2] Department of Computer Science, Faculty of Science, Kasetsart University
[3] The Graduate School, Kasetsart University

Abstract

Traditional attendance methods at The Graduate School, Kasetsart University can be time-consuming and susceptible to errors, while barcode systems have hardware needs and are susceptible to impersonation. To address these issues, we propose GradFace, an automated attendance system utilizing computer vision. GradFace uses a Gradio interface, employs the YOLOv11n-face model for real-time face detection, and leverages AWS Rekognition for accurate face identification. The system supports indexing faces from images linked to existing data and facilitates live registration via webcam. Experimental deployment during university events demonstrated stable performance and accurate face recognition across varied conditions (e.g., glasses, hairstyles, image quality), and positive feedback regarding convenience from 273 attendees. While network dependency and hardware requirements were noted as areas for improvement, GradFace successfully streamlined the registration process, generating attendance records and timestamped images with a latency of 1-2 seconds. Future work aims to enhance scalability, improve data management, explore local recognition.

Keywords: Face Recognition, Attendance Registration, Computer Vision, AWS Rekognition, YOLO

** Corresponding author: sarach.t@ku.th*

Received: *17 June 2025* ***Revised:*** *26 August 2025* ***Accepted:*** *11 September 2025*

1. Introduction

The Graduate School, Kasetsart University serves as a central hub for postgraduate education, regularly hosting a multitude of academic events, seminars, workshops, and orientations. These events often attract a large number of attendees, necessitating an efficient and reliable method for attendance registration. Traditionally, the Graduate School has relied on manual registration processes, typically involving attendees signing paper-based lists. However, this method is inherently inefficient, requiring significant workforce allocation for preparation, supervision, and data entry, consuming valuable time during event check-in, and being susceptible to human error or illegible entries. The increasing scale of educational institutions has made these methods increasingly inefficient.

In an attempt to improve efficiency and address the limitations of manual methods, the Graduate School explored various automated attendance systems. Quantitative data from prior manual registration processes reveals that an average event with 100 attendees often resulted in queues and data entry taking minutes, with errors from illegible handwriting or missed entries. Barcode and QR code systems, while faster, still required scanner deployment and were susceptible to proxy attendance, highlighting the need for a more robust and efficient

Existing research showcases various approaches to building such systems. System frameworks like Gradio and Streamlit facilitate the development of cross-platform applications with centralized server logic and accessible client interfaces (Abid et al., 2019). For the core task of face detection, techniques have evolved from traditional image processing methods like Haar cascades (Viola & Jones, 2001) to more robust deep learning models such as the You Only Look Once (YOLO) family (Redmon et al., 2016), which offer improved accuracy and speed. Other face detection algorithms include Multi-Task Cascaded Convolutional Neural Network (MTCNN) (Zhang et al., 2016), known for high accuracy.

Similarly, face recognition has progressed from methods like Scale-Invariant Feature Transform (SIFT) (Lowe, 2004) to sophisticated deep learning models that generate unique face embeddings for comparison, as well as powerful cloud-based services like AWS Rekognition (Indla, 2021). Deep learning models, particularly Convolutional Neural Networks (CNNs), have revolutionized face recognition by learning highly discriminative face embeddings, with models like InsightFace (Ren et al., 2023) achieving state-of-the-art performance and robustness. Online face recognition services, such as AWS Rekognition, provide pre-trained functionalities accessible via APIs, offering ease of use, scalability, and cost-effectiveness, though requiring careful consideration of privacy. Leveraging these technological advancements, we identified an opportunity to develop a more efficient, reliable, and user-friendly attendance system tailored to the needs of the Graduate School.

2. Objectives

This paper introduces GradFace, an automated attendance registration system designed and implemented specifically for the Graduate School at Kasetsart University. The primary goal is to streamline the event registration process by replacing manual and semi-automated methods with a robust face recognition solution. Explicit consent for data usage, including facial images and associated metadata, was obtained from all participating individuals prior to system deployment and data collection. Our contributions are as follows:

1. We developed an attendance registration system integrating a user-friendly web interface (Gradio), a real-time face detection module (YOLOv11n-face), and a cloud-based face recognition service (AWS Rekognition).

2. The system is designed to work with minimally modified existing operational data formats (Excel spreadsheets for metadata) and standard image files (PNG, JPG, WEBP) for facial data, ensuring ease of adoption.

3. We deployed GradFace in actual event scenarios at the Graduate School, Kasetsart University, evaluated its performance under operational conditions, and gathered feedback from both staff and attendees.

We classify GradFace as a face recognition system, as its primary function is to identify known individuals from a pre-indexed database, rather than simply verifying a one-to-one match. The choice of YOLOv11n-face for detection was based on its balance of speed and accuracy suitable for real-time webcam processing on the target hardware, while AWS Rekognition was selected for its proven accuracy, scalability, and ease of integration, minimizing local computational overhead for the recognition task.

3. Methods

3.1 System Architecture

GradFace is designed as a server-client application built primarily in Python, comprising several core components working together (see Figure 1).

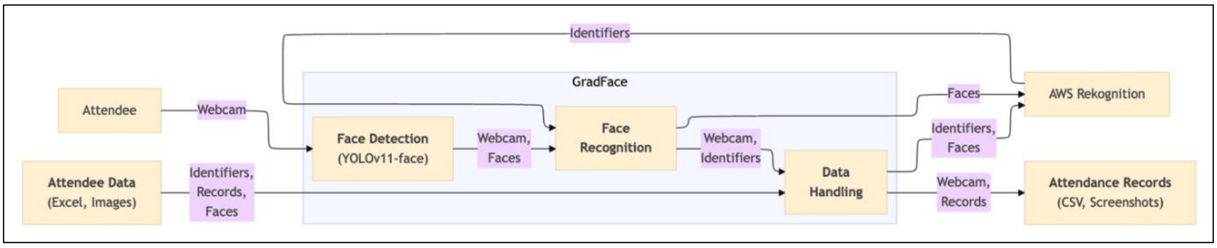

Figure 1 Overview of the GradFace system architecture, illustrating the flow of data and interactions between the attendee, server components, attendee data, and AWS Rekognition.

A key element is the system framework provided by Gradio, which delivers the web-based user interface, enabling cross-platform access via a browser and handling user interactions, data uploads, and real-time video streaming. Another crucial part involves data preparation and indexing, where the system processes attendee metadata from an Excel file and corresponding facial images, subsequently indexing the faces using AWS Rekognition to associate each unique identifier with specific facial features. For real-time operation, the system employs a face detection module based on YOLOv11n-face to analyze the live webcam feed and locate faces. Following detection, a face recognition component utilizes AWS Rekognition to compare the detected faces against the indexed collection and identify registered individuals. Lastly, the registration and output component handle the recording of attendance, displays relevant attendee information on the interface, and generates essential output files, including a Comma-Separated Values (CSV) log and timestamped screenshots for verification. The system also incorporates configurable parameters, such as detection sensitivity thresholds and recognition frequency, allowing for operational adjustments to optimize performance based on specific event requirements.

3.2 Interface Design

The user interface, built with Gradio (Abid et al., 2019), is structured for clarity and ease of use, organized into two main functional tabs, as shown in Figure 2 and Figure 3.

Register Tab: This tab (Figure 2) serves as the primary interface for conducting live attendance registration during events. This tab features a live webcam feed display, which is limited to an 800x600 resolution to maintain stable performance using WebRTC technology (Sredojev et al., 2015). Adjacent to the feed are controls allowing staff to adjust the sensitivity range for face detection. Users can also toggle the face recognition process on or off using a dedicated switch. A specific area within this tab is designated for displaying the photo and pertinent information, such as name and faculty, of successfully recognized attendees. For finalizing attendance, a manual "Register" button allows for staff confirmation, while an optional toggle switch enables automatic registration immediately upon positive identification. The system ensures that output files, namely the CSV log and webcam screenshots, are updated in real-time whenever a registration occurs.

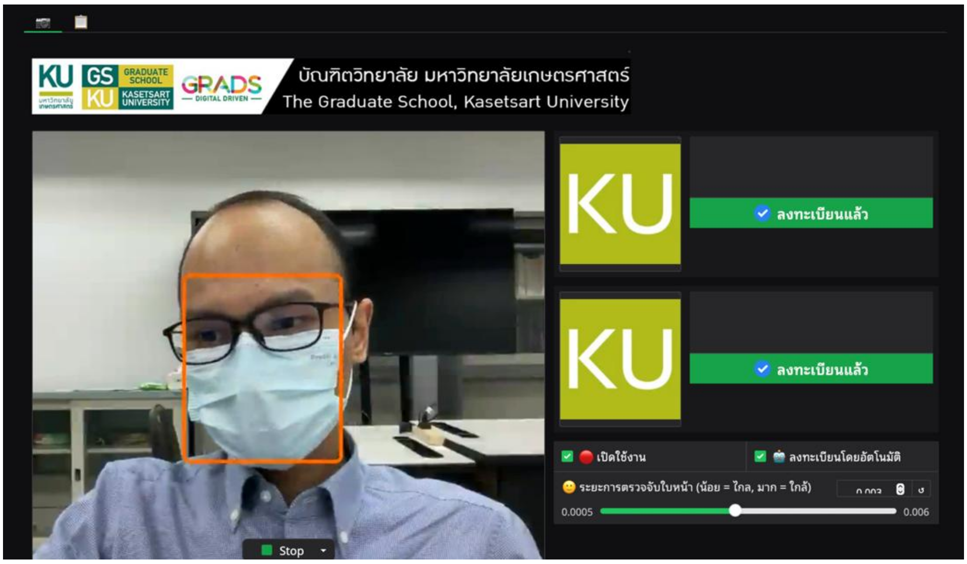

Figure 2 The Register Tab displays the live webcam feed, detection controls, and information for recognized attendees.

Upload Tab: This tab (Figure 3) is dedicated to data management tasks. Within this tab, users find controls for initiating the loading of metadata and face images, guided by paths specified in a configuration file. It also provides functionality for resetting the AWS Rekognition collection, effectively clearing all indexed faces, and for re-indexing faces from the currently loaded images into the AWS collection. These data handling processes are designed to be automated based on the predefined configuration settings.

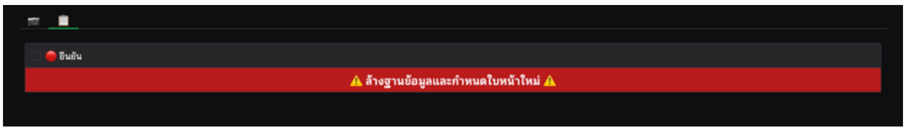

Figure 3 The Upload Tab provides controls for loading metadata and images and managing AWS Rekognition indexing.

3.3 Data Handling

To ensure seamless integration with existing operational procedures, GradFace is designed to utilize data formats already familiar to the Graduate School staff, necessitating only minimal adjustments. Attendee metadata is sourced from an Excel file, which contains essential information including a unique identifier (specifically constructed using the format FIRSTNAME_LASTNAME_NUM, where NUM is the image number), the individual's name prefix, faculty affiliation, and potentially other relevant details as required (Table 1).

Table 1 Key columns used for attendee metadata in the Excel file.

Column Name (Thai)	Description
คำนำหน้า	Prename/title (English and Thai)
ชื่อ	First name (English and Thai)
สกุล	Surname/last name (English and Thai)
คณะ	Faculty name (English and Thai)
สาขาวิชา	Major or field of study (Thai)
รหัสนิสิต	Student ID (or equivalent identifier)
ประเภท	Type (e.g., staff type, student type)

Complementing the metadata, face images are provided as individual files, with support for common formats like PNG, JPG, and WEBP. A critical requirement is that each image filename must precisely match the unique identifier found in the metadata file to establish the correct link. The system supports the use of multiple images per person, which enhances the robustness of the face recognition process. The indexing procedure is initiated from the Upload Tab during the data loading process. The system programmatically iterates through the metadata records and their associated images. For each person, their face image(s) are uploaded to the AWS Rekognition service and indexed using the corresponding unique identifier. This operation effectively builds a searchable collection of known faces within the AWS cloud service, ready for comparison during the live registration phase.

Recognizing the sensitive nature of biometric data, GradFace incorporates several technical privacy and security measures. All facial images are encrypted at rest using AWS S3 encryption standards (server-side encryption with S3-managed keys), and access to the AWS Rekognition collection is secured via AWS Identity and Access Management (IAM) roles with the principle of least privilege. Explicit consent for data usage, including facial images and associated metadata, was obtained from all participating individuals prior to system deployment and data collection via their student admission forms. This approach ensures that sensitive data is retained responsibly within the system during the events.

3.4 Face Detection

Real-time face detection is a critical component of GradFace, performed continuously on the incoming webcam stream. This is achieved using the YOLOv11n-face model, which was developed by GitHub user "akanametov" (akanametov, 2023) by fine-tuning an Ultralytics YOLOv11 nano model (Jocher et al., 2022) on the WIDER FACE dataset (Yang et al., 2016). To strike a balance between computational performance and detection accuracy on the designated hardware, the input resolution fed to the detection model is intentionally halved. Furthermore, the detection process is carefully managed with a target frame rate limit set at 10 frames per second (FPS). This limitation ensures stable, consistent operation and prevents excessive computational load on the system. As faces are detected in the video stream, they are visually highlighted by bounding boxes drawn directly onto the live webcam feed displayed within the Gradio interface, providing immediate visual feedback to the operator. The efficient transmission and display of the video stream are handled using WebRTC technology (Sredojev et al., 2015).

3.5 Face Recognition

When one or more faces are successfully detected by the YOLO model, the face recognition process is triggered, provided it has been enabled in the user interface. The high-level flow of this process is outlined in Algorithm 1.

Algorithm 1 The high-level flow for the face recognition and attendance registration process.

Input: Webcam Frame Stream, Known Faces Database F, Metadata M, Configuration (AutoRegister flag).
Output: UI Updates, Attendance Records (CSV, Screenshots).

Loop: Continuously process incoming frames.
  Capture Frame: Get current frame *I*.
  Detect Faces: Identify face regions *D* in *I* using YOLOv11n-face.
  For each detected face *f* in *D*:
    Recognize Face: Compare *f* against database *F* (using AWS Rekognition).
    If recognized match *id* found:
      Retrieve Info: Get attendee details Info(*id*) from *M*.
      Update UI: Display Info(*id*) and photo.
      Register (if applicable): If AutoRegister is True or manual trigger received, record attendance for *id*.
    Else (no match): Clear corresponding UI display area.
End Loop

In practice, the regions within the detected bounding boxes are first cropped from the current video frame. These cropped face images are then sent to the AWS Rekognition SearchFacesByImage API. To manage API costs and potential throttling, requests are batched, sending up to a default limit of 3 detected faces per second. AWS Rekognition compares the submitted faces against the pre-indexed collection (F) and returns the unique identifier (id) of the most likely matches. If a match is found, the system uses the returned identifier to look up the corresponding attendee's full information (name, faculty, image path, etc.) from the loaded metadata (M). This information, along with the attendee's indexed photo, is displayed in the Register Tab interface (Figure 1). If the auto-registration feature is enabled, or if the staff clicks the manual "Register" button, the attendance is logged in a CSV file with the current timestamp, and a screenshot of the webcam feed at that moment is saved for verification.

3.6 Evaluation Setup

The GradFace system underwent deployment and evaluation under authentic operational conditions during scheduled events organized by the Graduate School at Kasetsart University. The entire setup process and subsequent operation during the events were managed directly by the Graduate School staff, with the research authors providing necessary technical support and guidance.

Hardware: The configuration consisted of an MSI gaming laptop, which served as the host server running the GradFace application. This laptop was equipped with an NVIDIA GeForce RTX 2060 GPU to accelerate the deep learning computations. Video input was captured using a 4K resolution, wide-range external webcam connected to the laptop.

Software: The system operated within a Python 3.11 environment. GPU acceleration for the YOLO model was enabled through CUDA 12.6 and PyTorch 2.6.0. Essential libraries underpinning the system included Gradio, which provided the web interface framework, and Boto3, used for interacting with the AWS Rekognition API.

Dataset: The dataset used for indexing and testing is summarized in Table 2. Crucially, informed consent was obtained from all individuals involved, permitting the use of their images and associated data specifically for this system, prior to any participation.

Events: The events chosen for the primary evaluation included a significant orientation event for newly admitted students enrolling in the first semester of the 2025 academic year. To further assess performance and gather additional data, supplementary testing was also conducted during several smaller, functionally similar events hosted by the Graduate School.

Table 2 Summary of the experimental dataset used for GradFace evaluation.

Data Category	Description	Quantity/Details
Known Individuals	Graduate School Staff	31 people, 62 photos (DSLR, 2 poses/person)
Event Attendees	Actual Participants	242 people, 242 photos (Varied quality/poses)
Metadata	Attendee Information	Excel file (ID, Name, Faculty, etc.)
Consent	Data Usage Permission	Obtained from all participants

4. Results

The deployment and operation of GradFace during the selected events yielded encouraging results, affirming its potential as a viable automated attendance registration system. Key findings regarding system operation, performance, and feedback are summarized in Table 3.

Table 3 Summary of key evaluation results and user feedback from GradFace deployment.

Metric/Aspect	Finding	Notes
System Setup	Straightforward (Python env.)	Integrated with existing Excel workflows
Outputs	CSV Log (Timestamped), Screenshots	Generated consistently
Detection Performance	Stable 10 FPS (YOLOv11n-face); 1-2 false positives observed	Sensitivity adjustable, occasional over-detection
Recognition Accuracy	100% (AWS Rekognition)	Robust to variations (glasses, lighting etc.)

Metric/Aspect	Finding	Notes
Registration Time	~1-2 seconds per attendee	Significant improvement over manual (~15-30s) and QR (~5-10s) methods
Staff Feedback	Need guidance cues, Network/Hardware dependent	Positive overall
Attendee Feedback	Convenient, Faster, Modern	Positive overall

4.1 System Setup and Operation

The initial setup proved to be straightforward, primarily requiring the configuration of a standard Python environment and the installation of necessary library dependencies. The data import mechanism, designed to leverage existing Excel spreadsheet formats with minimal changes, significantly facilitated the integration of GradFace into the established operational workflows of the Graduate School staff. Throughout the events, the system consistently generated the expected outputs, namely a detailed CSV file logging all registered attendees along with their registration timestamps, and a corresponding set of webcam screenshots captured precisely at the moment each registration occurred, serving as visual verification.

4.2 Detection and Recognition Performance

Quantitative performance metrics observed during deployment demonstrate the system's reliability. GradFace achieved a stable 10 FPS processing rate for face detection with YOLOv11n-face on live webcam feeds, though 1-2 false positives (detecting non-faces as faces) were occasionally spotted. The face recognition component, leveraging AWS Rekognition, demonstrated 100% accuracy in identifying all registered attendees from the dataset during the experimental deployments. The average detection and recognition time, from frame capture to identified result display, was approximately 1-2 seconds per attendee. This significantly improved throughput compared to previous manual methods (which could take 15-30 seconds per attendee) or QR code systems (which could be 5-10 seconds per attendee). The system also demonstrated robustness by successfully identifying attendees across a wide range of challenging conditions. These included variations in ambient lighting, the presence of accessories like glasses, diverse hairstyles, significant variability in the quality of indexed photos (including older or lower-resolution images), and the presence of noise or artifacts in the input images.

4.3 User Feedback

Valuable staff feedback was collected during and after the events. Operational staff emphasized the need for clearer on-screen instructions or visual cues within the interface to better guide attendees on optimal positioning (e.g., where to stand or look) for efficient recognition by the camera. They also observed that the system's overall performance and responsiveness could sometimes be affected by fluctuations in network connectivity, particularly the latency involved in communicating with the AWS Rekognition service, and suggested that employing more powerful computing hardware could potentially enhance responsiveness. Attendee feedback was predominantly positive. Participants generally expressed appreciation for the convenience offered by the GradFace system, highlighting the elimination of the need to manually sign paper lists or remember to bring physical ID cards. The automated face recognition process was widely perceived as being significantly quicker and more modern compared to the traditional registration methods previously employed by the Graduate School.

5. Discussions and conclusions

The successful deployment and evaluation of GradFace clearly demonstrate the significant potential of integrating readily available deep learning models and cloud-based services to create effective, practical solutions for common operational challenges, such as event attendance registration. The strategic use of Gradio provided a simple yet highly functional interface, accessible universally via a standard web browser. Concurrently, the YOLOv11n-face model offered robust and efficient real-time face detection capabilities, while the reliance on AWS Rekognition ensured a high degree of face recognition accuracy, even under challenging real-world conditions. However, the evaluation also brought several areas for potential improvement into sharp focus.

5.1 Limitations and Challenges

The most prominent limitation identified was the architectural reliance on a single server instance, which was responsible for handling the entire workflow: processing the incoming webcam feed, executing the face detection model, managing recognition requests to the cloud service, and updating the user interface. As observed by the operational staff, network latency, particularly in communications with AWS, could introduce delays and impact the perceived speed of the recognition process. In scenarios where network stability might be compromised, this dependency could pose a significant operational challenge. This single-instance architecture also presents a significant scalability challenge, especially when considering larger events that might involve substantially more attendees, potentially numbering in the thousands (e.g., 2000+). For such scenarios, the current setup would likely prove insufficient. Hardware limitations also play a role, as more powerful computing resources could potentially improve responsiveness and allow for higher resolution or frame rate processing.

5.2 Data Management and Usability

While the current system achieves functional integration by utilizing external files like Excel spreadsheets and image folders, this approach relies heavily on manual file manipulation outside the application. Integrating more sophisticated and robust data management features directly within the GradFace application itself would significantly improve usability and streamline workflows. This could include built-in tools for importing data batches, exporting attendance logs in various formats, editing attendee records directly, and managing the associated image files, thereby reducing the dependency on external software and manual processes. Furthermore, usability could be enhanced by acting upon staff feedback regarding the need for clearer, intuitive on-screen visual guidance for attendees to help them position themselves correctly relative to the camera, optimizing the conditions for successful and rapid face detection and recognition.

5.3 Future Work

Addressing the limitations identified points towards several avenues for future work. A logical next step is to explore the implementation of a distributed system architecture, potentially incorporating multiple processing instances coordinated by a load balancer, to effectively handle the increased throughput demands of large-scale deployments and mitigate the single-server bottleneck. Enhancing data management capabilities within the application remains a priority for improved usability. Therefore, further investigation into optimizing user guidance within the interface and more rigorous performance evaluation with diverse metrics is warranted.

Another crucial aspect for future development involves refining the adaptability of the face detection model. Currently, YOLOv11n-face is hardcoded in Algorithm 1. While this provides stability for the current deployment, a more flexible approach could involve incorporating a mechanism for dynamic model selection or updates, allowing the system to easily integrate newer or better face detection models as they emerge, without requiring non-trivial code changes to the core detection logic.

Finally, a promising direction involves exploring the feasibility of utilizing local, on-device face recognition models. While potentially requiring more powerful local hardware, this approach could mitigate the system's dependency on network connectivity, potentially reduce recurring operational costs associated with cloud API calls, and importantly, enhance privacy by processing sensitive biometric data locally without transmitting it to external services. This presents a trade-off between initial hardware investment, ongoing costs, network reliance, and data privacy considerations.

Conclusion

This paper presented GradFace, a face recognition-based attendance system developed and implemented for the Graduate School at Kasetsart University. By effectively integrating a Gradio-powered web interface, the YOLOv11n-face model for real-time detection, and the AWS Rekognition service for accurate identification, the system successfully automated and streamlined the event registration process, replacing previous manual methods. Experimental deployment demonstrated GradFace's promising performance with its capability for accurate attendee identification across diverse and challenging conditions, receiving positive feedback from users regarding its convenience. While the evaluation highlighted challenges related to network dependency for cloud recognition and hardware requirements for optimal performance, GradFace successfully streamlined registration, generating automated attendance records and corresponding timestamped images. Future development efforts will focus on enhancing user guidance features, improving the system's scalability, and further investigating the potential advantages and trade-offs of employing local, on-device recognition models.

6. Acknowledgment

We would like to express our sincere gratitude to the Graduate School, Kasetsart University, for the opportunities provided and to their staff for their valuable cooperation, feedback, and assistance during the experimental deployment of the GradFace system. We also acknowledge the financial support provided by the Department of Computer Science, Faculty of Science, Kasetsart University.

7. References

Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., & Zou, J. (2019). *Gradio: Hassle□free sharing and testing of ML models in the wild* (arXiv:1906.02569) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1906.02569

akanametov. (2023). *Yolo-face* [Computer software]. GitHub. https://github.com/akanametov/yolo-face

Indla, R. K. (2021). *An overview on Amazon Rekognition technology* [Master's thesis, California State University, San Bernardino]. CSUSB ScholarWorks. https://scholarworks.lib.csusb.edu/etd/1263

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., Michael, K., TaoXie, Fang, J., imyhxy, Lorna, Zeng, Y., Wong, C., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, ... Jain, M. (2022). *ultralytics/yolov5: v7.0 – YOLOv5 SOTA realtime instance segmentation* (Version v7.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.7347926

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788. https://doi.org/10.1109/CVPR.2016.91

Ren, X., Lattas, A., Gecer, B., Deng, J., Ma, C., & Yang, X. (2023). Facial geometric detail recovery via implicit representation. *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*, 1-8. https://doi.org/10.1109/FG57933.2023.10042505

Sredojev, B., Samardzija, D., & Posarac, D. (2015). WebRTC technology overview and signaling solution design and implementation. *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1006-1009. https://doi.org/10.1109/MIPRO.2015.7160422

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001),* I-511–I-518. https://doi.org/10.1109/CVPR.2001.990517

Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). WIDER FACE: A face detection benchmark. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5525–5533. https://doi.org/10.1109/CVPR.2016.596

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters, 23*(10), 1499-1503. https://doi.org/10.1109/LSP.2016.2603342