PROJECT PROPOSAL

Date of proposal:
09.10.2024

Project Title:

Group ID (As Enrolled in Canvas Class Groups): Practice Project Group (FT) 5

Group Members (name , Student ID):

Gong Xinyu(A0295988W)
Liu Peilin(A0291951X)
Su Chang (A0296250A)
Wang Xinji(A0265810H)
Zhang Wenqing(A0296886Y)

Sponsor/Client: *(Company Name, Address and Contact Name, Email, if any)*

None

Background/Aims/Objectives:

1 Social Background

Globally, there are around 70 million deaf people. Many of them use various forms of sign language as their primary means of communication. The World Health Organization (WHO) estimates that 430 million people worldwide currently experience disabling hearing loss, a figure expected to rise to 700 million by 2050. Deafness and hearing loss significantly impact education, employment, and social inclusion, especially in developing countries. In Singapore, it's estimated that there are 500,000 individuals with hearing loss. Of these, only 5,400 are registered with the Singapore Association for the Deaf (SA Deaf), which offers various services such as sign language courses and assistive technologies.

2 Problems encountered by deaf people

Despite improvements in the living conditions of deaf people in modern society and the gradual increase in social inclusiveness towards people with disabilities, communication barriers between ordinary people and deaf people are still evident. This stems mainly from the fact that most ordinary people lack the ability to sign, resulting in communication methods that are limited to typing or relying on text interpreters. These methods are cumbersome and inefficient in many situations, especially when quick and effective communication is required. For example, in emergency situations or in the service sector, the inability to apply sign language fluently often delays the delivery of information.
We believe that to build a truly harmonious society, in addition to providing convenient infrastructure, it is also necessary for members of society to enhance their understanding and tolerance of people with disabilities.

Sign language is an important tool for the hearing impaired to express themselves. If more people can master the basic sign language alphabet, the communication efficiency and tolerance of the society will be greatly enhanced.

3 Aim of system

By developing a mobile app, we aim to help ordinary people to quickly master the basic sign language alphabet so that they can communicate with the hearing impaired in a simple but effective way. We combine AI model and computer vision to do the gesture recognition. The app is divided into two main modes: **Learning mode** and **Practice mode based on word spelling**. Through automatically recognizing if the user's gesture matches the letter shown, users can master the sign language alphabet in a short period of time with high accuracy.

We expect that with the continuous optimization of the software and enhanced cooperation with various public welfare organizations, more ordinary people will be able to master the basic sign language alphabets through this platform in the future and apply this knowledge in their daily lives to help the hearing impaired. This will help reduce communication barriers in society, promote understanding and cooperation between people, and contribute to the harmonious development of society.

Project Descriptions:

1 Project Scope

1.1 Resources Requirement

- **Hardware proposed for consideration:**
  - Local/cloud system with RTX 3090 GPU
- **Software proposed for consideration:**
  - Front end:
    - Android Mobile Phone
  - Middleware:
    - Flask
  - Back end:
    - Relational DB: SQLite or MySQL
    - Pre-trained Model: YOLOv5
  - Programming Language:
    - Python
    - SQL
    - Java
    - C++

1.2 Use Case Scope:

Use Case ID	**UC 01a**
Use Case Name	Basic Sign Language Alphabet Learning
Description	The user can browse all the sign language of the alphabet and the number list, and practice sign language in front of the camera. The system provides real-time feedback on the user's gestures, helping them learn and improve their sign language skills interactively.
Preconditions	• The user is on the "Learning" page, which also serves as the home screen for the system. • The camera is enabled, and gesture recognition is active.
Trigger	The user selects the "**Alphabet Learning**" option on the "Learning" page to enter the alphabet learning module.
Basic Flow	1. The system displays the complete alphabet, each letter paired with some corresponding images of the sign language gesture. 2. The user selects a letter or a number, and the system prepares to capture the user's gesture. 3. The user makes the corresponding sign language gesture in front of the camera. 4. The system analyzes the gesture using computer vision or gesture recognition algorithms and provides feedback on accuracy or corrections if needed. **5.** The user can proceed to the next letter or continue practicing the current one until they master it.

Alternative Flows	1a. The user can add this letter/number to their favorite list, which can be seen at **UC 03** "Me" page. 4a. The system calculates the confidence level for each gesture. If the confidence score is greater than 0.8 (a threshold that can be adjusted during development), the gesture is accepted as valid. 4b. Correct gestures are marked as "pass" for each letter.
Postconditions	The user returns to the previous page.
Special Requirement	• The system should save practice records, tracking completed words and showing the user's progress, which can also be seen at **UC 03** "Me" page. • Real-time feedback must be supported through fast and accurate gesture recognition. • The interface should be user-friendly, with accurate gesture recognition even under varying lighting conditions and camera angles.
Business Rules	Silent learning mode: Both the system and user can remain silent during the exercise.
Open Issues	Investigate the feasibility of integrating 3D modeling for a more immersive learning experience. This could allow users to rotate and view sign language gestures from various angles, enhancing understanding.

Use Case ID	**UC 01b**
Use Case Name	Basic Sign Language Word/Numbers Testing (Optional)
Description	The user practices signing a combination of letters and numbers, such as English words or phone numbers, using sign language gestures in front of the camera. The system analyzes these gestures and provides feedback after the user completes the string, supporting interactive learning and improving the user's sign language proficiency.
Preconditions	• The user is on the "Learning" page, which serves as the home screen for the system. • The camera is enabled, and gesture recognition is active.
Trigger	The user selects the "**Word/Numbers Testing**" option on the "Learning" page to begin practicing with randomized letter and number strings.
Basic Flow	1. The system generates a random string (letters and/or numbers) and displays it alongside an input field. 2. The user signs the corresponding gesture for each character in front of the camera.

	3. The system analyzes each gesture using computer vision or gesture recognition algorithms and captures the input if it meets the accuracy threshold. 4. The user may proceed to the next character (of this string) or choose to delete the previous character and try again. 5. After completing the string, the user taps the "Submit" button to check their answers. 6. The system evaluates the entire string, providing feedback on accuracy or corrections if needed.
Alternative Flows	1a. The user can also add this word/number string to their favorite list, which can be seen at **UC 03** "Me" page. 3a. The system calculates the confidence level for each gesture. If the confidence score is greater than 0.8 (a threshold that can be adjusted during development), the gesture is accepted as valid. 3b. Correct gestures are marked as "pass" for each word.
Postconditions	The user returns to the previous page.
Special Requirement	• The system should save practice records, tracking completed words and showing the user's progress, which can also be seen at **UC 03** "Me" page. • Real-time feedback must be supported through fast and accurate gesture recognition. • The interface should be user-friendly, with accurate gesture recognition even under varying lighting conditions and camera angles.
Business Rules	Silent learning mode: Both the system and user should remain silent during the exercise.
Open Issues	This is an optional use case, as it has lots of overlapping places with **UC 02**.

Use Case ID	**UC 02**
Use Case Name	Sign Language Game
Description	The user plays a game where words or number strings slowly fall from the top of the screen. To clear each falling word before it reaches the bottom, the user must sign the correct corresponding letter or number in front of the camera. The system uses sign language gesture recognition to analyze and match the user's input, shooting down the word if the gesture is correct.
Preconditions	• The user is on the "**Game**" page. • The camera is enabled, and gesture recognition is active.
Trigger	The game starts when the user selects the "Start Game" button.

Basic Flow	1. The system generates random words or numbers, which start falling from the top of the screen. 2. The user must make the correct sign language gestures for each character in front of the camera. 3. The system processes each gesture using gesture recognition algorithms, checking if the user's gesture matches the current character in the falling word. 4. If the gesture is correct, the corresponding character in the word is "shot down," and the user moves on to the next character. 5. The user continues this process until all characters in the falling word are cleared before reaching the bottom of the screen. 6. Once all characters of the word are cleared, the user earns points and proceeds to the next word or level.
Alternative Flows	3a. The system calculates the confidence of gesture recognition. If the confidence score is greater than 0.8, the character is successfully shot down. 5a. If the user fails to sign the correct gesture before the word reaches the bottom, the system penalizes the user (e.g., reducing points or health). 6a. As the game progresses, the speed of falling words increases, adding more challenge to higher levels.
Postconditions	The user returns to the previous page or a game over screen after all levels are completed or if the user loses the game.
Special Requirement	• The system must provide real-time feedback through fast and accurate gesture recognition. • The game should become progressively more difficult as the user advances in levels, with words falling faster and containing more complex characters (symbols, punctuation, etc.). • The game should track user scores and progress, showing achievements or levels unlocked. The progress should be saved and can be seen at **UC03** "Me" page.
Business Rules	Silent learning mode: Both the system and user should remain silent during the exercise.
Open Issues	Investigate potential enhancements, such as power-ups or bonuses for clearing multiple words in quick succession, or offering a leaderboard for competitive play.

Use Case ID	**UC 03**
Use Case Name	Me Page (Profile)
Description	The user accesses the "Me Page" (Profile) to view personalized information, such as their saved letters, words, and numbers for

	future practice, along with tracking their learning progress and viewing game records, including highest scores. The page acts as a central hub for users to monitor their overall performance and quickly access favorite or challenging learning items.
Preconditions	• The user is logged into the application. (Optional) • The system has stored the user's learning and game history.
Trigger	The user navigates to the "Me Page" from the home screen or other parts of the application.
Basic Flow	1. The system loads the "Me Page" upon user request. 2. The system displays the following sections: **Review List**: A list of letters, words, numbers, or symbols that the user has saved for future review. **Learning Progress**: A summary of how many words, letters, or numbers the user has learned, e.g., "You have learned 50 words." **Game Records**: The user's highest score in the sign language-based game and other key game statistics, such as completed levels or average score per session. 3. The user can select any saved item from the **Review List** section to begin reviewing it again. 4. The user can view detailed learning stats, such as the total time spent learning or a breakdown of their progress by category (letters, words, numbers). 5. The user can view and compare their game records, including high scores and recent achievements. 6. The user can navigate back to the home page or other sections from the "Me Page."
Alternative Flows	3a. If the user has not saved any items for reviewing, the system displays a message like "You have not saved any items yet." 5a. If the user has not played any games yet, the game records section shows a message like "No game records available."
Postconditions	The user can return to the previous page or continue exploring different sections within the "Me Page."
Special Requirement	• The page should load quickly and present information in a visually organized manner, allowing easy navigation between favorites, learning progress, and game records. • The system should allow users to remove items from their **Review List** and update learning progress in real-time.
Business Rules	The system should only show items and progress related to the currently logged-in user. (Optional)

Open Issues	Investigate the possibility of adding login functions so the aforementioned “optional” functions can be achieved.

2 Data Collection and Preparation

2.1 Data Collection

To build a diverse and high-quality sign language recognition dataset, we plan to collect data through multiple channels, ensuring the model's ability to generalize across different scenarios. The steps are as follows:

- **Public Dataset Acquisition:**

We have obtained 720 sign language gesture images from public datasets, which cover all characters of the sign language alphabet. These datasets, sourced from widely recognized sign language resources, provide the foundational data for our model training.

- **Custom Data Collection:**

In addition to using public datasets, our team members plan to take additional sign language gesture images to further expand the dataset. Team members will simulate different use cases, including various lighting conditions, gesture angles, and background settings, to increase the diversity of the data. We anticipate collecting more types of sign language gestures, enhancing the model's ability to perform well in complex real-world scenarios.

- **Data Collection Goal:**

By combining public data with custom data, we aim to create a rich and diverse dataset that covers all characters of the sign language alphabet. The dataset will include gestures taken from various angles, lighting, and backgrounds, ensuring robustness in different scenarios.

2.2 Data Annotation and Augmentation

Once the initial data collection is complete, data annotation and augmentation are critical steps to ensure the effective training of the sign language recognition model. We will accurately annotate each sign language image and use data augmentation techniques to expand the dataset.

- **Gesture Annotation:**

We will use tools like LabelImg to manually annotate each image, ensuring that each sign language gesture is correctly marked with a bounding box. Accurate annotation is essential for improving the model’s recognition performance. Through precise manual annotation, we ensure that the model can accurately identify and locate the gesture region during training.

- **Data Augmentation:**

To enhance the model’s generalization ability, we will apply data augmentation techniques to the existing dataset. These techniques include:

Rotation: Rotating images at different angles to simulate gestures from various perspectives.

Horizontal and Vertical Flipping: Increasing the diversity of gestures by flipping the images, preventing the model from becoming overly dependent on gestures in a single direction.

Brightness and Contrast Adjustment: Simulating gestures under different lighting conditions to improve the model's adaptability to low-light and bright environments.
Blurring: Applying blur to some images to improve the model's ability to recognize gestures in unclear or low-resolution images.

- **Dataset Expansion:**

By applying these data augmentation techniques, we will significantly expand the original dataset, ensuring the model can handle a variety of scenarios and gesture variations. This diversity will greatly improve the model's performance in real-world applications, especially when faced with changes in lighting, angles, or backgrounds.

2.3 Data Processing

To ensure the sign language images are suitable for training deep learning models, we will perform standardization and preprocessing on the images, ensuring consistency and efficiency during the training process.

Image Standardization:
All images will be resized to a unified resolution of 1024×1024 pixels, allowing the model to handle a consistent input format. This standardization ensures efficient training and robust model performance during testing and real-world application.

Data Preprocessing:
After resizing the images, we will normalize the pixel values of all images to the [0,1] range. This normalization step accelerates the model's training convergence and prevents issues like gradient explosion or gradient vanishing during the training process.

3 System Design

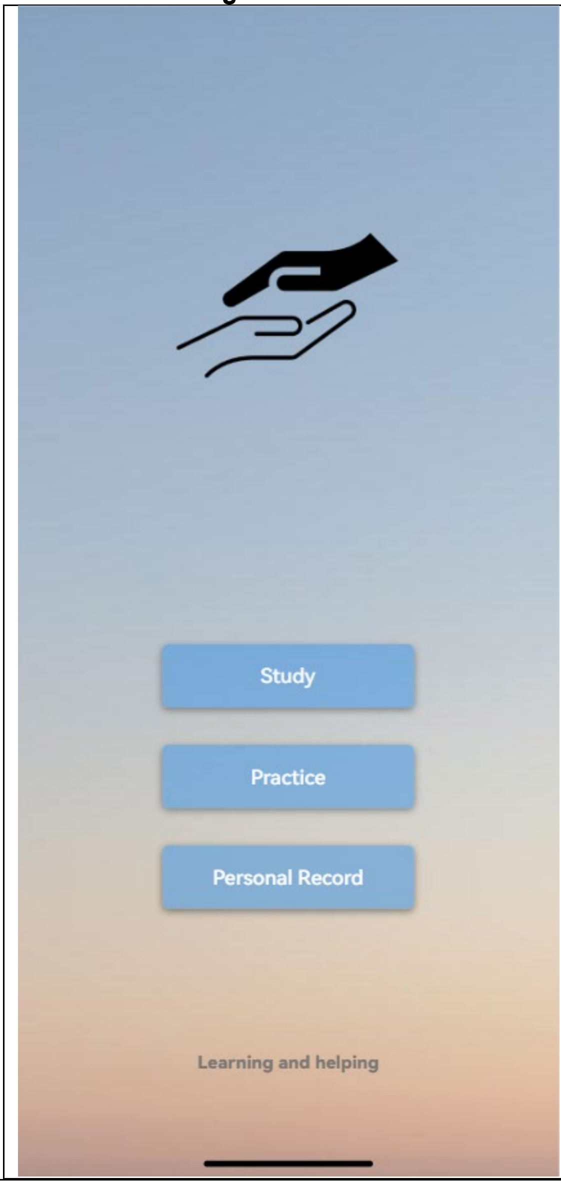
3.1 model method

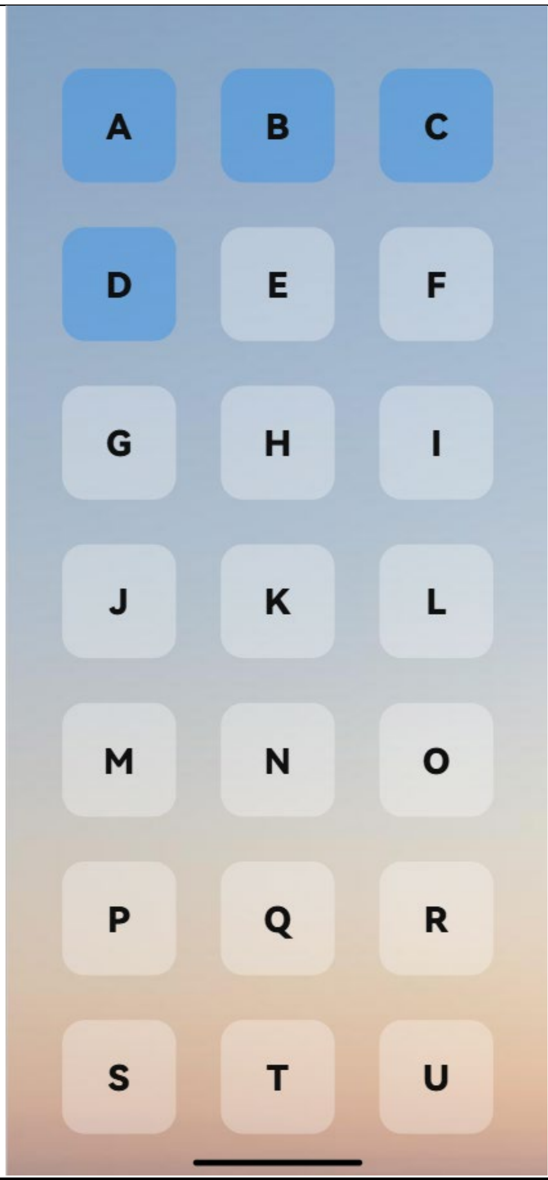
- **Model Selection and Motivation:**
  Chose the YOLOv5 model for sign language recognition mainly because of its efficiency in real-time object detection and low computational requirements, making it ideal for mobile devices. YOLOv5 is flexible, supports transfer learning, and can be quickly adapted to specific tasks. Its high accuracy and low latency ensure that users receive fast feedback and a smooth experience.
- **Data Preprocessing**:
  Data Collection and Labeling: Used a public dataset of 720 images of sign language gestures, which were manually labeled using LabelImg. Each image's gesture was marked with a bounding box to ensure the model could accurately identify the gesture.
  Data Augmentation: To improve the model's generalization,applied techniques such as image rotation, blurring, horizontal flipping, and color adjustments. This increased the dataset to 18,000 images, helping the model perform well in various scenarios.
- **Model Architecture and Training:**
  YOLOv5 Architecture: YOLOv5 consists of a backbone network, feature pyramid, and detection head. We used the YOLOv5m version, utilizing transfer learning with a pre-trained model to speed up fine-tuning on the sign language dataset.
  Training Process: Training was conducted on 1024×1024 resolution images, with a batch size of 8 for 300 epochs. The final mAP@0.5 reached 98.17%, and the mAP@0.5 to 0.95 range was 85.27%.
- **Inference and System Integration:**
  Real-Time Inference: The system captures gestures through the camera in real-time and uses YOLOv5 to recognize them. If the confidence score is above 0.8, the system considers the gesture correct and provides immediate feedback.

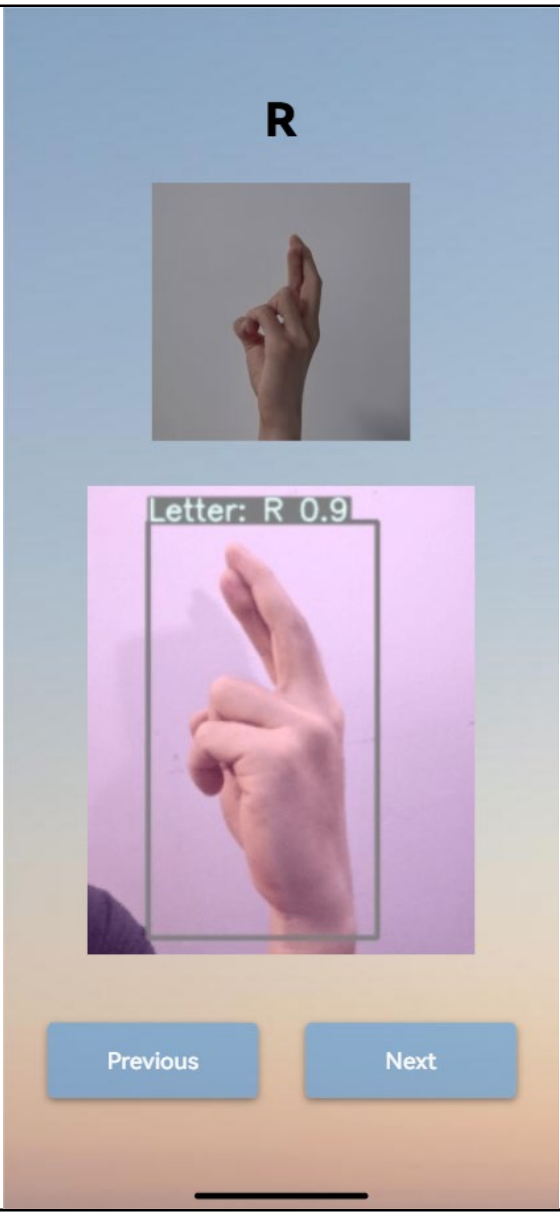
System Optimization: We optimized inference speed and reduced image resolution to keep the system's response time in the millisecond range. The frame rate is maintained above 30 FPS, ensuring smooth sign language recognition in different environments.

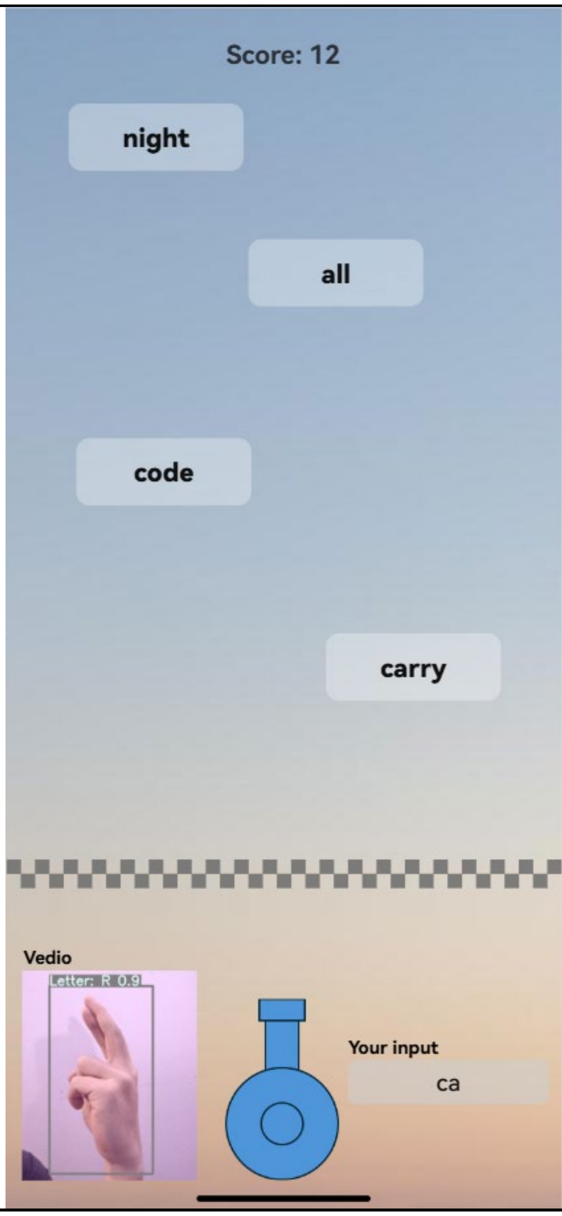
- **System Performance and Robustness:**
  The system was tested in various conditions, including low light and different angles. The response time stayed within 50 milliseconds, ensuring both real-time performance and robustness, delivering a consistent sign language recognition experience.

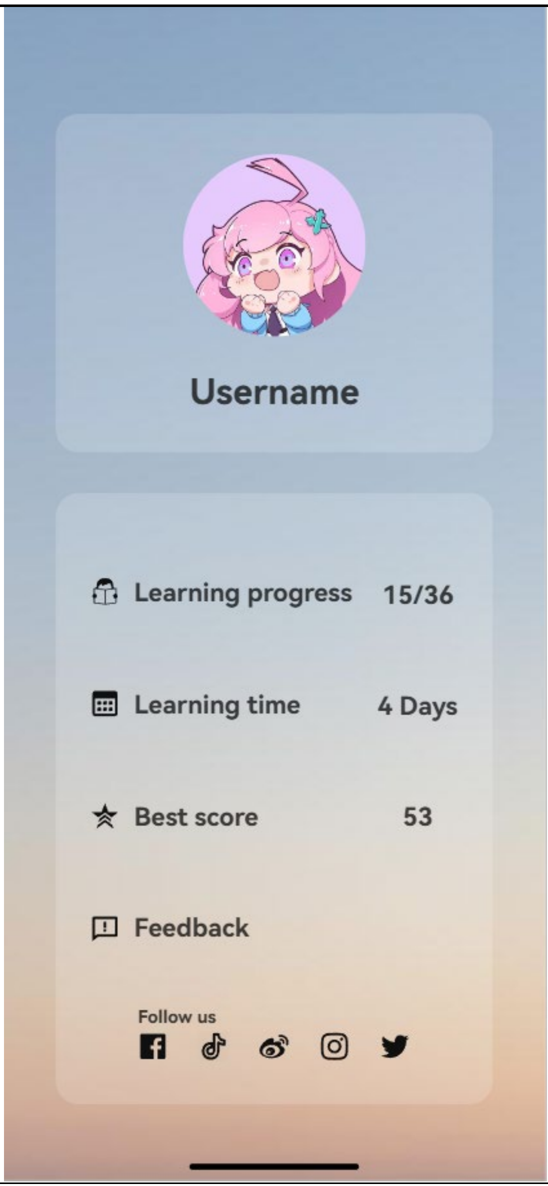
3.2 UI design






R
Letter: R 0.9
Previous
Next


Score: 12
night
all
code
carry
Vedio
Letter: R 0.9
Your input
ca



4 **Evaluation**

4.1 Accuracy Evaluation

- **Accuracy**
  Definition: Calculate the proportion of gestures correctly recognized by the model to the total recognized gestures.
  Task: Evaluate the accuracy of the application's recognition of user gestures through a test set to ensure that users can quickly master sign letters.
  Formula:
  Accuracy = (Number of correctly recognized gestures/total gestures) × 100%
- **Precision and Recall**
  Precision: The proportion of all predicted gestures that are actually correct
  Recall rate: In all cases where it is actually a gesture, the prediction is the proportion of that gesture
  Task: Evaluate the model's ability to recognize different gestures in both learning and practice modes, and help optimize the gesture library.

Formula:
Precision = (true case/(true case + false positive case)) × 100%
Recall rate = (true case/(true case + false negative case)) × 100%

- **F1 score**
  Definition: An indicator that combines accuracy and recall
  Task: Balance accuracy and recall in learning and practice models, ensuring that the model is neither too strict to miss gestures nor too loose to misidentify gestures.
  Formula: F1 = 2 × (accuracy × recall rate)/(accuracy + recall rate)

4.2 Classification Performance Analysis

- **Confusion Matrix**
  Definition: Shows the model's predictions for each category
  Task: Help us to identify which gestures the model performs well on and which are often confused so that targeted improvements can be made.

4.3 Real-time Evaluation

- **Latency and frame rate**
  Definition: Evaluate the model's response time and frames per second (FPS) processed in real-time gesture recognition
  Task: Ensure that gesture recognition is responsive enough to improve the user experience when using the app.
  Metrics: Response time (milliseconds) and frame Rate (FPS)

4.4 Model Robustness

- **Performance under different circumstances**
  Definition: Tests the recognition ability of the model under different lighting conditions, backgrounds, and variations of user gestures.
  Task: Ensure that the model works effectively in a variety of real-world use scenarios

4.5 Baseline Comparison

- **Comparison with other models**
  Definition: Evaluate the relative performance of this model compared to similar gesture recognition models.
  Task: Understanding the competitiveness of our model to drive further optimization.
  Metrics: Comparative accuracy, recall and F1 scores.