REAL-TIME SIGN GESTURE RECOGNITION USING AN ADAPTABLE CONVOLUTIONAL NEURAL NETWORK MODEL

Introduction:

Technological advancement in communication is essential in today's society to ensure equal media and service access for everyone, including those with disabilities. The deaf and hard-of-hearing population face significant communication challenges in an oral language-dominated society. Current technologies provide limited support for sign language. Our project, "LSL SignBot," aims to bridge this gap by employing a Convolutional Neural Network (CNN) model based on MobileNetV2 to recognize sign gestures in real-time. This system serves the needs of the deaf and hard-of-hearing community by utilizing deep learning and computer vision concepts (Alashhab et al., 2019; Zhao et al., 2018).

Background and Rationale:

Despite advancements in communication technologies, the deaf and hard-of-hearing population still face barriers due to the dominance of spoken language. Sign language, their primary means of interaction, is underrepresented in digital technologies (Ren et al., 2021). The proposed "LSL SignBot" project addresses this challenge by employing a flexible CNN model based on MobileNetV2 to recognize real-time sign gestures (Liu et al., 2019). This system enhances communication for the deaf and hard-of-hearing population, integrating them more fully into the modern technological world (Mahesh et al., 2017).

There are existing mobile applications for hand gesture recognition, such as those utilizing deep convolutional networks for tiny hand gestures (Bao et al., 2017) and mmWave sensors for mobile devices (Ren et al., 2021). However, our application distinguishes itself by focusing on real-time sign gesture recognition specifically for the deaf and hard-of-hearing community, leveraging MobileNetV2 for enhanced accuracy and efficiency. This app aims to provide a more accessible, user-friendly, and efficient solution compared to current offerings.

Research Questions:

1. What are the difficulties in recognizing different sign languages?
2. What strategies can be adopted to enhance real-time sign gesture recognition on mobile devices?

3. Which preprocessing techniques should be applied to increase the recognition rate of sign language data?

These research questions indicate a combination of qualitative and quantitative research. We plan to conduct online surveys and questionnaires to understand the challenges of recognizing different sign languages. Additionally, we will focus on developing and testing the mobile application to address these questions.

Methodology:

Theoretical Resources: Based on deep learning, computer vision, and sign language recognition literature (Alashhab et al., 2019; Bao et al., 2017).

Research Approach: Implement a CNN modified for real-time sign gesture recognition. The CNN structure will be based on MobileNetV2, which has been previously used for similar applications (Mahesh et al., 2017).

Research Methods:

- **Data Gathering and Cleaning:** Collect a variety of sign gestures from participants familiar with sign languages. Ethical approval will be sought for this data collection. Pre-process the data to remove errors and inconsistencies using standard data cleaning techniques (Liu et al., 2019).
- **Image Preprocessing:** Resize, normalize, convert to grayscale, reduce noise, and enhance contrast. Apply ROI cropping to extract relevant parts of the image (Bao et al., 2017). Tools such as OpenCV and Python libraries will be used for this preprocessing.
- **Region Extraction:** Use background subtraction, skin color detection, motion detection, and hand detection models to pinpoint areas in images and videos (Zhao et al., 2018; Ren et al., 2021).
- **Feature Extraction:** Utilize Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), CNN features, landmark points, and temporal features for dynamic signals (Liu et al., 2019).

- **Model Configuration and Training:** Tailor a MobileNetV2 architecture CNN and fine-tune it with augmented training data (Mahesh et al., 2017). TensorFlow and Keras will be used for model development.
- **Real-Time Processing:** Implement OpenCV for real-time video processing, detection of hands, and tracking within the mobile application. Ensure acceptable performance concerning computational load, memory, power consumption, and battery life (Zhao et al., 2018).
- **User Interaction Design:** Develop user-friendly prompts and feedback, and design an interface for the app (Alashhab et al., 2019). The design strategy will involve user-centered design principles.
- **Performance Evaluation:** Assess model accuracy, precision, and recall. Conduct comprehensive functional tests using tools like scikit-learn and evaluate the performance in various scenarios (Bao et al., 2017).
- **Ethical Considerations:** Ensure user data security and consent. Perform usability tests with proficient sign language users and incorporate their feedback (Ren et al., 2021).
- **Impact Assessment:** Evaluate the system's effectiveness in enhancing the quality of life for the deaf and hard-of-hearing population by facilitating communication and reducing social exclusion (Mahesh et al., 2017).

Plan of Work and Time Schedule:

Given the timeline, the project phases have been adjusted to ensure feasibility within the available time frame:

- **Initial Phase (Week 1-2):**
    - Literature review
    - Data collection
    - Data set compilation (Bao et al., 2017)

- **Development Phase (Week 3-4):**
    - Data preprocessing
    - Model setup
    - Initial training (Liu et al., 2019)
- **Testing Phase (Week 5-6):**
    - Further model assessment
    - Performance checking
    - Real-time information processing testing (Ren et al., 2021)
- **Implementation Phase (Week 7):**
    - Interface design
    - Interface testing
    - User feedback method development (Mahesh et al., 2017)
- **Final Phase (Week 8):**
    - User feedback collection
    - Final system and component performance assessment
    - Documentation (Alashhab et al., 2019)

References:

Alashhab, S., Gallego, A. J., & Lozano, M. Á. (2019). Hand gesture detection with convolutional neural networks. In *Distributed Computing and Artificial Intelligence 15th International Conference 15* (pp. 45-52). Springer International Publishing.

Ren, Y., Lu, J., Beletchi, A., Huang, Y., Karmanov, I., Fontijne, D., & Xu, H. (2021, March). Hand gesture recognition using 802.11 ad mmWave sensor in the mobile device. In *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)* (pp. 1-6). IEEE.

Bao, P., Maqueda, A. I., del-Blanco, C. R., & García, N. (2017). Tiny hand gesture recognition without localization via a deep convolutional network. *IEEE Transactions on Consumer Electronics, 63*(3), 251-257.

Zhao, H., Ma, Y., Wang, S., Watson, A., & Zhou, G. (2018). MobiGesture: Mobility-aware hand gesture recognition for healthcare. *Smart Health, 9*, 129-143.

Liu, P., Li, X., Cui, H., Li, S., & Yuan, Y. (2019). Hand gesture recognition based on single-shot multibox detector deep learning. *Mobile Information Systems, 2019*, 1-7.

Mahesh, M., Jayaprakash, A., & Geetha, M. (2017, September). Sign language translator for mobile platforms. In *2017 International Conference on Advances in Computing, Communications, and Informatics (ICACCI)* (pp. 1176-1181). IEEE.