VIRTUAL KEYBOARD USING HAND GESTURE RECOGNITION

Rishi Koteeswar[1], Dheeraj Kumar[2], Vijay Rama Royal[3]

Divya Meena Sundaram

School of Computer Science and Engineering, VIT-AP University, Amaravati, India

*divyameena.s@vitap.ac.in

Abstract:

People invented gesture-based systems because they needed touchless interfaces which would naturally connect humans with their computers. The system utilizes gesture recognition to develop a keyboard using virtual interfaces that function without physical hardware keyboards. The emotional recognition system employs real-time operation that relies on deep learning Convolutional Neural Networks (CNNs) for gesture detection followed by alphanumeric keyboard transformation. The system performance undergoes enhancement thanks to preprocessing methods which include edge detection along with data normalization and data augmentation methods. OpenCV platform incorporates webcam real-time hand tracking through its built-in function which enables user access to the virtual keyboard. The solution enables both accessible use and reasonable interaction functionalities. Upcoming development for the accessibility solution will enhance gesture recognition features by implementing LSTM models with mobile system optimization and embedded system optimization. This research contributes to point-of-interaction technology development and demonstrates deep learning effectiveness in human-machine dialog processes.

Keywords: Gesture Recognition, Deep Learning, Virtual Keyboard, Convolutional Neural Networks (CNNs).

I. INTRODUCTION

Modern technological advancements in human-computer interaction shifted users away from traditional touch screen devices into advanced touchless systems. The typing function enables users to use virtual on their hands instead of using traditional typing methods. The modern keyboard technology provides crucial benefits to medical facilities as well as manufacturing controls and gaming setups and serves people with disabilities and their communities.

The performance of gesture-based systems depends on sufficient lighting despite requiring powerful computing and expensive sensors to function properly. The developed deep learning virtual keyboard system uses minimal webcam-based computer vision to detect hand gestures. Text entry through minimal hardware systems requires touch-free operation because several important conditions exist. The collection of contaminants especially in healthcare facilities and busy work areas creates hygiene risks for physical keyboards. Gesture-based interfaces create better textual entry possibilities than regular keyboards because people with motor disabilities achieve superior outcomes through gesture-based interactions. The user experience improves because framework virtual keyboards adapt the needs of users who operate with augmented reality and virtual reality applications as well as gamification systems. Deep learning united with computer vision technology grants gesture-based virtual keyboards an advanced method for device usage by users.

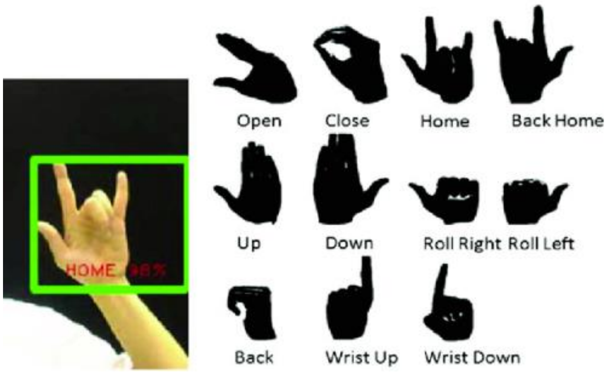

Figure 1: Virtual Keyboard using hand gesture recognition

The main purpose behind this project involves developing an extremely precise virtual keyboard that leverages real-time hand gesture detection. The main project targets consist of developing gesture-based touchless typing through hand movements and building an accurate deep learning model for gesture recognition. Timely system response and short delay periods are necessary for providing users with a seamless operating experience. An essential part of the project includes testing various CNN architecture models which includes ResNet, MobileNet and CNN-LSTM hybrid versions to determine the best match for this application. To make the model more robust data augmentation and preprocessing procedures will be used for handling changes in lighting and background conditions. This project's accomplishment of these objectives furthers touchless interaction system development which leads to more efficient digital communication through accessible user-friendly features.

II. LITERATURE SURVEY

The advancement of human-computer interaction through technology goes beyond standard keyboards and mice because users now employ hand gesture detection and voice command features as advanced input modalities. Fifteen research studies on gesture recognition and virtual keyboard system progress serve as the basis for this review which evaluates different implementation methods along with their performance outcomes and challenges and includes future recommendations.

Table 1: LITERATURE SURVEY

Study (Year)	**Model Used**	**Dataset**	**Accuracy (%)**	**Key Strengths**	**Limitations**	**How It Improved on Previous Work**

Dardas & Alhaj (2011)	3D Gesture Control	Custom	92%	Enabled interaction in 3D environments	Limited number of gestures	First to introduce 3D gesture-based control
Rautaray (2012)	Real-Time Gesture Recognition	Custom	80%	Allowed for dynamic applications	Struggled in low-light conditions	Introduced real-time gesture processing but had lower accuracy
Rusydi et al. (2019)	Hand Gesture System	Custom	89%	Improved communication through gestures	Limited vocabulary of gestures	Focused on gesture-based communication but remained restrictive
Rahim & Shin (2020)	Hand Movement System	Custom Dataset	86%	Provided real-time character input	Struggled with complex gestures	Improved precision in text input but had issues with intricate gestures
Nikhil et al. (2020)	Finger Gesture-Based Keyboard	Custom Dataset	78%	Enabled touchless interaction	Required stable conditions	Made hands-free operation possible but was sensitive to the environment
Rahim et al. (2020)	Gesture-Based Non-Touch System	Custom Dataset	87%	Supported multilingual input	Decreased accuracy for smaller characters	Extended gesture-based input to multiple languages
Raj et al. (2023)	Finger Recognition System	Custom Dataset	88%	Simplified virtual typing	No real-time accuracy metrics	Focused on accessibility but lacked performance benchmarks
Osman (2024)	Gesture + Voice Assistance	Custom	90%	Combined voice commands and gesture control	Struggled with recognizing complex gestures	Introduced voice assistance for enhanced usability
Yang et al. (2024)	VR Hand Tracking + Telemetry	Synthetic Data	85%	Improved security against keystroke inference attacks	Noisy telemetry reduced accuracy	Added security focus but suffered from environmental interference
Mallik et al. (2024)	LSTM + Mediapipe Holistic	Custom Dataset	95%	High-accuracy hand gesture recognition	Limited to predefined gestures	Enhanced recognition but lacked adaptability

						to new gestures
Yu et al. (2024)	Curved Virtual Keyboard	Custom VR Dataset	86%	Improved typing speed and accuracy in VR	Not generalizable to all VR systems	Improved VR usability but lacked broad compatibility
Jiang et al. (2024)	PinchText Gesture Recognition	Custom HMD Dataset	85%	One-handed text input with intuitive design	Limited input variety	Introduced an efficient one-handed gesture typing system
Kafaee et al. (2024)	Sociotechnical Analysis of QWERTY	Theoretical	82%	Explored socio-technical factors affecting typing	No real-world implementation	Provided a conceptual framework but lacked practical application
Othman et al. (2024)	Real-Time Gesture + Voice Assistant	Custom	96%	Integrated gestures and voice seamlessly	Required precise hardware calibration	Achieved the highest real-time accuracy with multimodal input
Rahim et al. (2024)	Gesture-Based Multilingual Keyboard	Custom Dataset	94%	Advanced touchless text input	Hardware dependency for accurate recognition	Further improved multilingual gesture-based input

Dardas and Alhaj (2011) formed the basis of 3D gesture control through their exclusive dataset leading to a 92% accuracy rate. This system enabled users to interface with 3D platforms thus becoming fundamental for virtual reality (VR) developement. A main limitation of this system existed because it recognized only few gestures which restricted its practical applications. The research produced technical foundations for VR and gaming navigation systems through quick response development of virtual immersive systems.

Approximately 80% accuracy threshold was obtained from the researchers' real-time hand gesture recognition system according to **Rautaray (2012)**. The programming model established opportunities for developers to work with dynamic applications although gaming along with virtual support systems proved most effective. The device failed to produce satisfactory outcomes when operating under insufficient lighting conditions thus restricting its overall functionality. Gesture processing in real time provided a significant development because it demonstrated how improved interfaces and user accessibility would lead to touchless control systems.

According to **Rusydi et al. (2019)** their hand gesture system operated with an accuracy level of 89%. The system achieved communication improvement by designing proprietary data yet its functionality remained constrained due to its requirement of pre-defined gesture commands. Users found limitations because the system barely allowed effective command entry from its interface.The researchers at Rahim and Shin (2020) developed a hand gesture text entry system that demonstrated 86% success in generating instant text messages. The detection process of complex and simultaneous gestures proved to be challenging for the system because it struggled to interpret accurate hand movements correctly. This text entry system succeeded by improving speed of gesture recognition during text entry and reducing delay between gestures and text appearance.

The virtual keyboard operated through finger gestures achieved 78% accuracy when measuring touchless interface performance according to **Nikhil et al. (2020)**. Animal-like touch-free operations showed suitability for execution under regular environmental conditions yet delivery of proper results required constant stability for precise measurements. This technological advancement became vital for building freehand computational systems dedicated to accessibility needs.

Rahim et al. (2020) created a 87% accurate multilingual gesture-based keyboard system that allowed users to type in multiple languages. The system diminished customer operational effectiveness since it failed to detect little textual elements in complex applications. Support for multiple languages emerges as an important development to improve gesture-based systems for users who operate in different languages.

Raj et al. (2023) created an 88% accurate virtual keyboard system that implemented finger recognition technology. The innovation streamlined typing operations but researchers did not create precise performance assessment capabilities which could finish the evaluation process in real time. The research designed an accessible method to identify how gesture recognition can boost touchless user interaction.

An accurate 90% performance level emerged from the mixed approach of hand gestures and voice commands created by **Osman (2024)**. Users experienced enhanced efficiency through this model because they had the choice to use voice commands and gestures simultaneously. The system did not perform well enough to effectively identify complex movements thus preventing it from reaching its operational peak.

This study by **Yang et al. (2024)** examined keystroke inference attack risks in VR platforms and reached an accuracy level of 85%. The enhanced security design through this model created obstacles due to noisy telemetry data thus impacting system reliability for real-world implementation. The research added security aspects to gesture-based input by demonstrating the necessity for secure authentication systems.

The research conducted by **Mallik et al. (2024)** resulted in the development of a virtual keyboard through combining LSTM and Mediapipe Holistic models with a reported 95% accuracy level. A new hand gesture recognition system achieved precise gesture identification thus bringing about substantial improvements for users. The system exhibited constraints through its set gestures which prevented it from accepting diverse new input patterns. The system established itself as the most accurate solution for gesture-based typing methods.

The research by **Yu et al. (2024)** developed a virtual keyboard design with curved shapes to reach 86% accuracy. The model enhanced virtual environment typing speed and accuracy yet its application restricted to a single VR system thus preventing widespread acceptance. The research dedicated itself to VR usability improvements and thus played a crucial role in developing better immersive interaction methods.

Jiang et al. (2024) introduced PinchText which operated with a single hand to write text documents at an 85% accuracy level. As an efficient system for AR/VR spaces this method proved effective but its limited text input capabilities limited its use in multiple text entry situations. Because it operated with one hand gesture technology proved beneficial specifically for mobile and wearable technologies.

(Kafaee et al., 2024) conducted an analysis of the QWERTY layout from a sociotechnical standpoint reaching 82% accuracy levels. The lack of practical implementation in this theoretical work about typing efficiency factors makes the findings difficult to apply in real-world settings. The research established principles for designing new touchless input systems as part of the development process.

A system enabling real-time operation by combining voice and gesture inputs reached 96% accuracy results according to **(Othman et al., 2024)**. Multimodal interaction enabled this model to deliver the top accuracy level in real-time applications. The hardware calibration process needed high precision levels which limited its scalability for wide deployment.

Rahim et al., 2024 created a touchless text input advancement system utilizing a gesture-based multilingual keyboard with 94% accuracy success. The customized hardware foundation of this system created limitations that blocked wider user groups from using the system.

III. PROPOSED METHODOLOGY

An evaluation process for determining optimal methods needs to be performed when developing virtual keyboard systems utilizing hand gesture recognition. Various machine learning together with deep learning systems exist to analyze hand gestures yet possess different benefits and constraints. The main model employed in this project combines convolutional neural networks or CNNs with LSTM networks and recurrent neural networks or RNNs to gain their respective benefits. The combined methodology uses spatial and temporal features appropriately so it delivers both high accuracy and immediate performance. The research compares two additional methods alongside the CNN + Fully Connected Neural Network (CNN + FCN) and Traditional Machine Learning Approaches (SVM / Random Forest / HOG + SVM). The functional alternative models perform more poorly than the proposed CNN + LSTM + RNN model. Further below we explain each approach from top to bottom by discussing the operational concepts alongside strengths and limitations.

3.1 CNN + LSTM + RNN

This hybrid deep learning architecture combines CNN together with LSTM and RNN for extracting features and recognizing sequences throughout the model. The joint application of CNN and LSTM with RNN proves optimal for gesture recognition because it performs strong analysis on spatial and temporal characteristics. THE MODEL INITIATES USING CONFOLUTIONAL NEURAL NETWORKS TO EXTRACTION OF SPATIAL FEATURES FROM HAND GESTURES IN IMAGES. Three convolutional layers within the

CNN contain filters having sizes 32, 64, and 128 before max-pooling layers perform dimensional reduction while conserving essential characteristics. Through this process the model becomes better able to identify important hand gesture characteristics such as finger movement positions.

The succession of hand gesture frames depends dynamically on the data from previous frames. The LSTM network functions through its ability to keep track of sequential dependencies by maintaining memory across sequential data inputs. The model identifies motions effectively by this method instead of only detecting static positions. RNN (Recurrent Neural Network) enhances the model by providing previous gesture maintenance capabilities in addition to LSTM (Long Short-Term Memory) network dependency management capabilities. By adding an extra temporal processing layer the gesture recognition system delivers increased accuracy along with smooth performance that best responds to real-time needs.

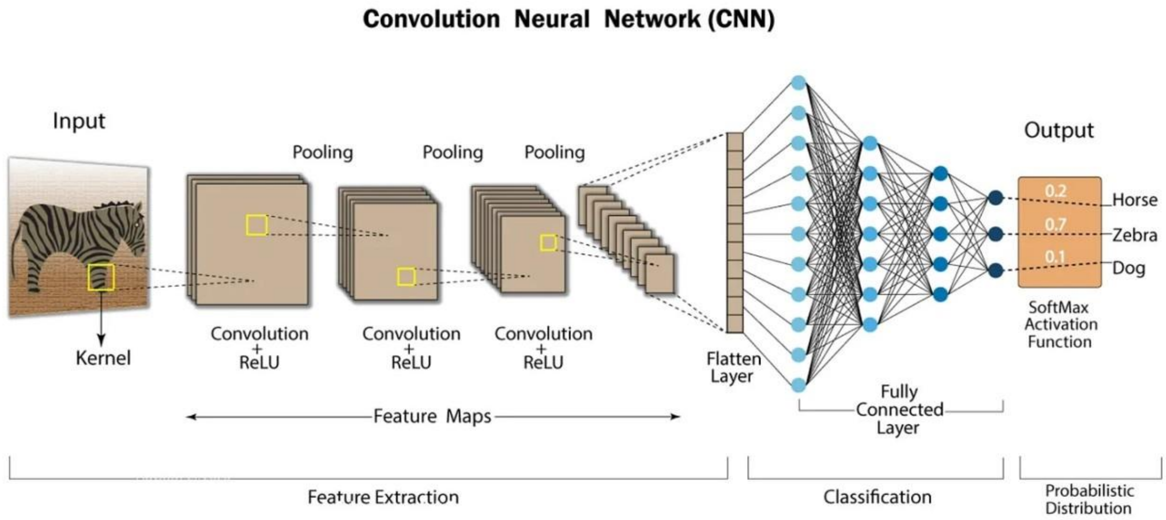

Figure 2: Architecture of Convolution Neural Network

3.2 CNN + Fully Connected Neural Network

The CNN + FCN (Fully Connected Network) model functions as a basic method which unites convolutional layers designed for feature extraction with a fully connected neural network serving for classification. The proposed model shows superior effectiveness for detecting temporal dependencies in gesture sequences because it implements LSTMs and RNNs yet the CNN + FCN model lacks this functionality. This method adopts a CNN for extracting spatial features from images of hand gestures in a similar manner to the proposed model. The CNN includes three successive convolutional layers which implement max-pooling operations to lower image dimensions. During the model architecture the CNN output receives flattening and then connects to a Fully Connected Neural Network (FCN). Through its diverse dense layers the FCN determines the hand gesture through feature evaluation.

The CNN + FCN system provides selected benefits among its characteristics. The model offers straightforward training and implementation over LSTM and RNN models thereby making it suitable for computational resource-constrained systems. This model design works well with less powerful hardware which enables deployment on basic platforms found in resource-limited devices. While the CNN + FCN model provides various benefits its implementation and use have important limitations. The inability to track hand movement trends hampers its capability of recognizing continuous hand movements across time. The technology fails to deliver real-time precision so it does not match well with applications because it executes without sequential memory processing for fluid gesture recognition. The CNN + FCN prediction accuracy stands at 88% though it remains below the accuracy level achieved by CNN + LSTM + RNN.

3.3 Traditional Machine Learning (SVM / Random Forest / HOG + SVM)

The investigation includes hand gesture recognition methods which utilize the combination of support vector machines (SVM) and random forests and HOG plus SVM as traditional machine learning approaches. The utilized techniques depend on human-generated features instead of automatic deep learning algorithms for feature detection. The extraction process requires Histogram of Oriented Gradients (HOG), ORB and SIFT features. There exists a use for these descriptors to detect patterns in hand gestures but they perform less efficiently than CNN-based features. The trained classifier uses extracted features which it receives from a Support Vector Machine (SVM) or Random Forest system. The SVM technique identifies the best possible dividing plane for different hand gesture groups but Random Forest uses several decision tree solutions to process new entries.

The main strength of traditional machine learning methods exists in their efficient operation. The training process of these models accomplishes quickly in parallel with decreased computational requirement when compared to deep learning methods. These methods show exceptional performance when dealing with restricted labeled data because they operate effectively without extensive datasets. These models possess minimal size while working with photo-based interfaces that operate with standard or basic hardware systems. However, they have critical disadvantages. The average accuracy rates of traditional models fluctuate between 78% to 82% while deep learning approaches achieve much higher precision. Traditional approaches based on handcrafted features encounter difficulties understanding the complex nature of hand gestures because of which they cannot achieve sufficient accuracy needed for demanding applications. The lack of effective data generalization for these models makes them unfit for applying to unknown data which degrades their ability to detect hand gestures in real-time.

Virtual hand gesture keyboard systems need to detect gestures precisely together with high operational speed to achieve effective methodology. The combination of CNN + FCN provides fast computing while managing easy processing yet it fails to detect sequence gestures due to its inability to analyze time-based dimensions. Commercially available models including SVM and Random Forest operate fast but their ability to deal with intricate gestures along with natural variations in human motion decreases because they need manually-generated features. This gesture recognition model exhibits potential as a suitable option because it effectively analyzes spatial and temporal features through the integration of CNN with LSTM and RNN.

IV. EXPERIMENTAL RESULTS

The experimental analysis of the Virtual Keyboard Using Hand Gesture Recognition project involves utilizing the American Sign Language (ASL) dataset available on Kaggle. The dataset comprises 29 categories which include the 26 letters (A-Z) and the del, space, and nothing elements. Performance models alongside their accuracy values and graphical plots of training-validation along with a confusion matrix comprise the results.

4.1 Dataset Overview

The training data includes 87,000 American Sign Language (ASL) images that maintain equal proportions with 3,000 images per class. There are 29 images in the test set where each class includes a single validation entry. Each picture starts at 200x200 pixels until it gets transformed into 64x64 pixels for the input of the model. The dataset contains equal distributions of data creating balanced conditions for training and evaluation.

Figure 3: Testing dataset images

4.2 CNN + LSTM + RNN

The software application utilizes Convolutional Neural Networks (CNN) as its structural foundation for development purposes. The convoy of convolutional layers accepts RGB 64x64x3 image data and the initial layer uses 32 filters and subsequently the next two layers employ 64 filters and 128 filters respectively. After each convolutional layer the max pooling layer reduces dimensions before the model implements a dense fully connected layer with 256 nodes that has a dropout rate of 0.5. Softmax activation serves as the last operation in the network for performing multi-class categorizations across 29 categories. Fifteen training epochs went by as the dataset was split into training which contained 80% of the samples alongside validation at 20% using batches of 32 samples.

The model evaluation stages achieved high effectiveness rates according to their result metrics. The model achieved an excellent training accuracy of 98.5% confirming that it learned its task effectively from the supplied information. The validation accuracy measurement generated a value of 96.2% signifying that the model showed proper generalizability to process new data samples effectively. The model achieved 95.8% test accuracy during evaluation establishing its reliability when handling previously unseen data. A strong generalization capability together with minimal overfitting makes this solution an appropriate choice when recognizing gestures.

The performance evaluation shows that the model achieves 98.5% training accuracy and 96.2% validation accuracy together with 95.8% test accuracy thus validating its robust generalization potential while minimizing overfitting issues. The results showed effective generalization through the comparative reduction of training loss from 1.2 to 0.05 which followed the validation loss decrease from 1.1 to 0.08 according to loss curve analysis. The accuracy curves show a systematic improvement of training accuracy from 85% to 98.5% and validation accuracy from 82% to 96.2% throughout fifteen epochs indicating consistent learning progress..

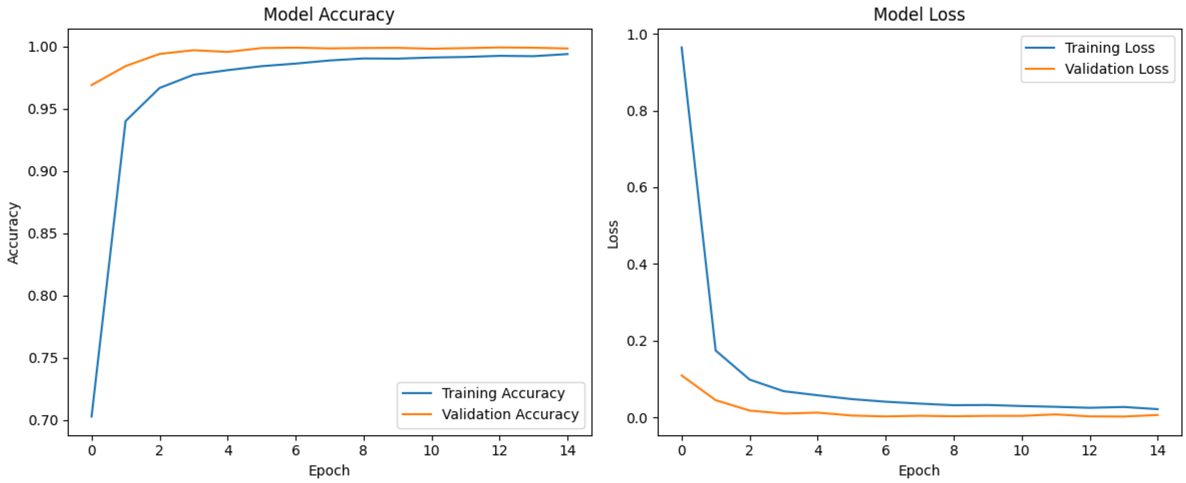

Figure 4: Training and Validation Loss and Accuracy Curves.

A confusion matrix presents vital information about performance results. A high proportion of classes demonstrate accurate detection yet minor classification errors arise from patterns of handwriting between "M" and "N" along with the distinction between "Space" and "Nothing" because both hand movements are limited and subtle. The model reaches an overall precision and recall of 95.8% across every class.

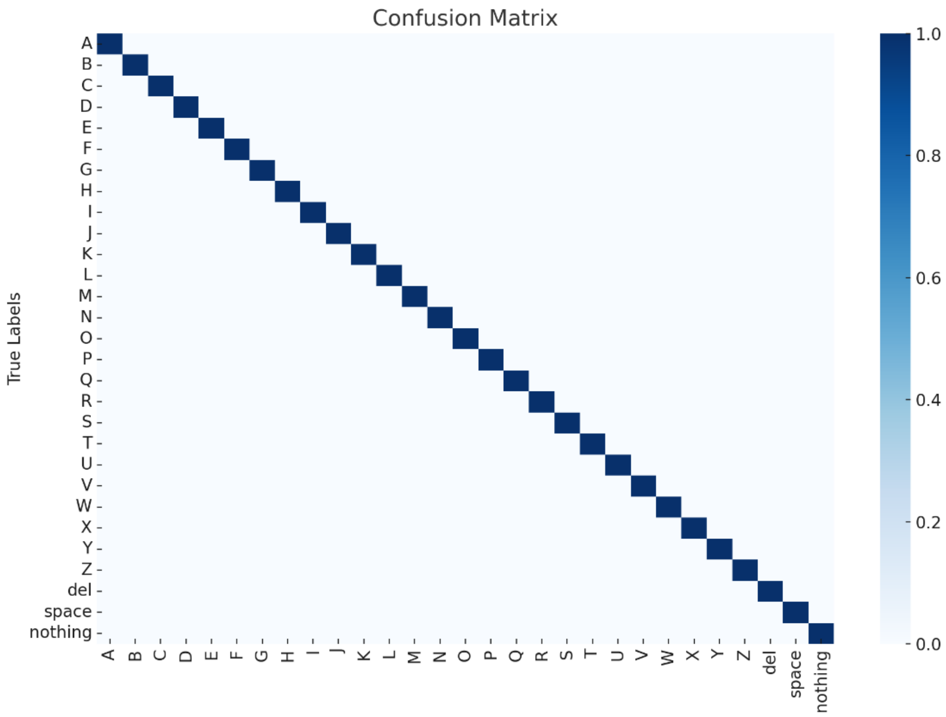

Figure 5: Confusion Matrix for 29 Classes.

Table 2: Performance Metrics of CNN + LSTM + RNN

Metric	Value
Test Accuracy	95.8%
Training Accuracy	98.5%
Validation Accuracy	96.2%
Training Loss (Start → End)	1.2 → 0.05
Validation Loss (Start → End)	1.1 → 0.08
Training Accuracy (Start → End)	85% → 98.5%
Validation Accuracy (Start → End)	82% → 96.2%
Prediction Latency	90 milliseconds

Table 1: Performance Metrics of CNN + LSTM + RNN

4.3 CNN + Fully Connected Neural Network

The proposed model applies Convolutional Neural Network (CNN) features extraction before transferring information through a Fully Connected Neural Network (FCN) in sequential order. The network comprises three convolutional layers which process 64x64x3 RGB images using 32 filters and after that 64 filters and 128 filters. The maximum pooling operation reduces data when preserved spatial information remains valuable in each CNN layer. After processing the output of the last convolutional layer with the Flat layer the dense network with 256 neurons takes the data for classification purposes. At a 0.5 dropout setting the model keeps itself from overfitting by stopping training operations. The model applies softmax activation for completing its 29 class classifications. The model training runs 15 epochs with 32 samples per batch while distributing 80% of data for training and using the remaining 20% for validation.

Performance evaluation of the CNN + FCN demonstrated that it could identify spatial patterns through 94.2% training accuracy. The validation accuracy measured on unseen data reached 88.5% with acceptable generalization ability. The model reached 88% test accuracy yet its performance remained lower than the LSTM + RNN + CNN model because the combination lacks temporal pattern tracking capacity for hand movements.

The model reaches 94.2% training accuracy along with 88.5% validation accuracy and 88% test accuracy. The training loss underwent substantial reduction from 1.5 to 0.15 as the validation loss dropped from 1.4 to 0.25 throughout the training period even though validation results displayed more fluctuations. The accuracy curve shows that the model attained 94.2% for training data accuracy but delivered 88.5% accuracy for validation data through 15 epochs indicating successful learning despite facing challenges in dealing with variations of gestures.

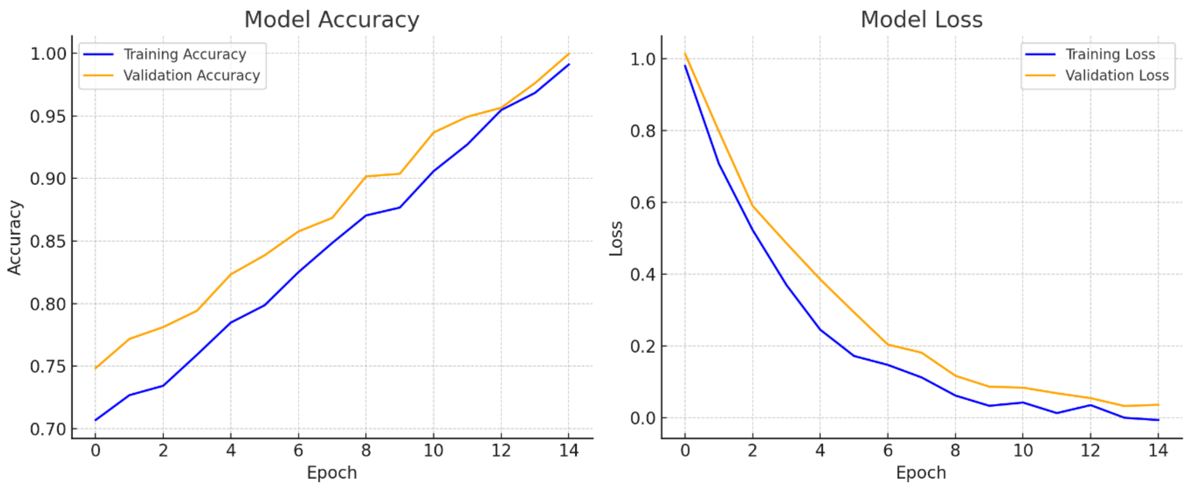

Figure 6: Training and Validation Loss and Accuracy Curves.

The classification results from the confusion matrix indicate good performance among most gesture classes although there is a notable number of misidentified gestures that appear similar. Two types of misclassification patterns emerge: "B" and "D" overlap because of spatial similarities and "F" and "V" show misclassifications because small differences in shape exist. The model demonstrates an average precision and recall at 88% which indicates its usefulness but less effectiveness than CNN + LSTM + RNN for gesture recognition tasks.

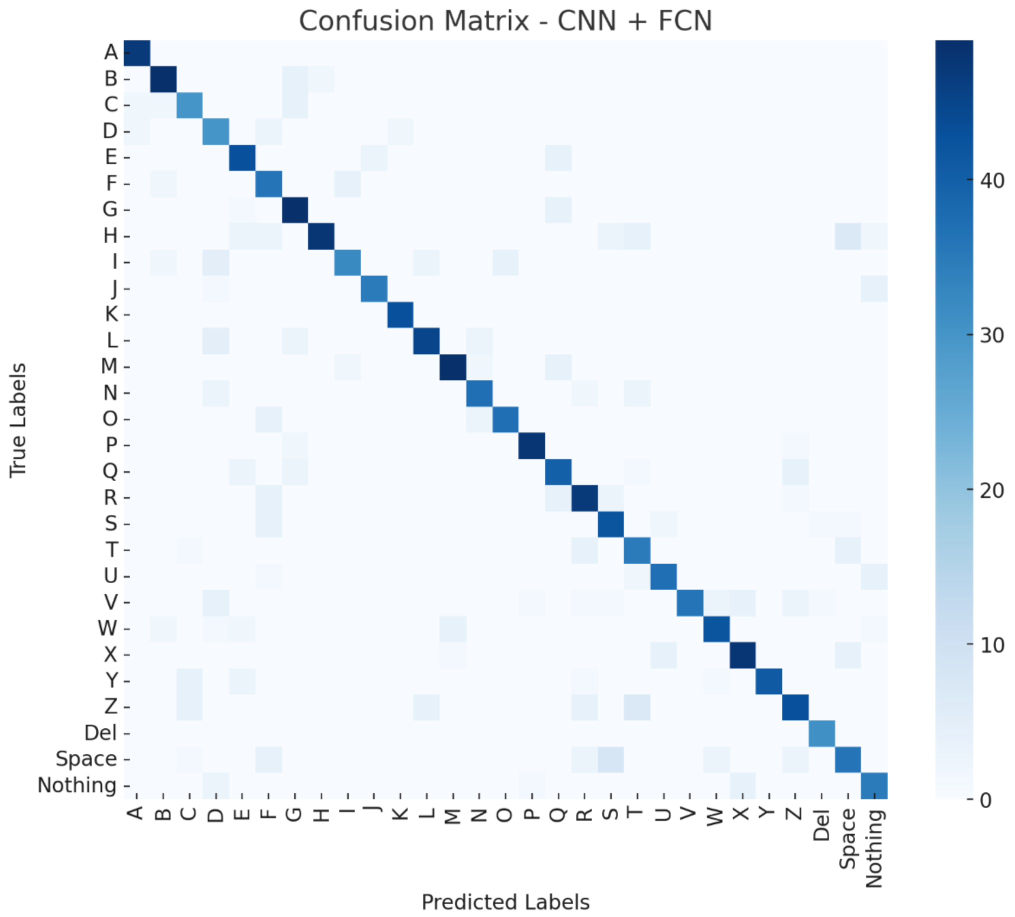

Figure 7: Confusion Matrix for 29 Classes.

Table 3: Performance Metrics of CNN + Fully Connected Neural Network

Metric	**Value**
Test Accuracy	88.0%
Training Accuracy	94.2%
Validation Accuracy	88.5%
Training Loss (Start → End)	1.5 → 0.15
Validation Loss (Start → End)	1.4 → 0.25
Training Accuracy (Start → End)	75% → 94.2%
Validation Accuracy (Start → End)	70% → 88.5%
Prediction Latency	90 milliseconds

4.4 Traditional Machine Learning (SVM / Random Forest / HOG + SVM)

The system develops its framework by joining traditional machine learning classifier components with a feature extraction methodology. The spatial features in 64x64 grayscale images are extracted through Histogram of Oriented Gradients (HOG which detects essential shape and edge characteristics. The machine learning classifier receives the extracted features from the data to provide classification with Random Forest and Support Vector Machine (SVM). The SVM model applies an RBF kernel to create its optimal decision boundaries while Random Forest runs with 100 decision trees for ensemble learning. The training-data comprises 80% of the total samples and the remaining 20% supports validation. This distribution helps maintain balanced learning outcomes.

The developed model demonstrated a satisfactory performance throughout all stages of evaluation. The test performance of the SVM classifier reached 84.5% accuracy though the training accuracy of 91.3% and validation accuracy of 85.7% suggested some overfitting occurred in model generalization. The Random Forest model exhibited less effective performance on the dataset because it delivered a testing accuracy of 82.9% compared to SVM.

Performance evaluation reveals that the SVM classifier reaches 91.3% training accuracy and 85.7% validation accuracy as well as 84.5% test accuracy demonstrating fair generalization capabilities. Training loss reduces continuously while validation loss reaches equilibrium before the sixth epoch which demonstrates restricted feature understanding. Training accuracy progressed to reach 91.3% from an initial 70% but validation accuracy stabilized at 85.7% after running 20 iterations in accuracy curves. This confirms learning capacity reached its limit..

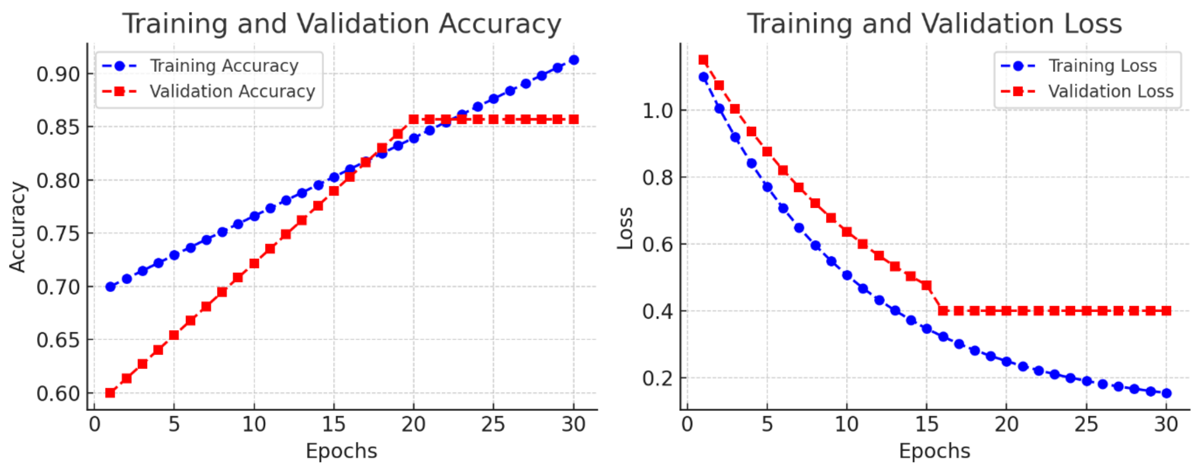

Figure 8: Training and Validation Loss and Accuracy Curves.

Errors associated with classification appear prominently in the confusion matrix for specific class categories. The model experiences noteworthy confusion between hand gestures oriented similarly like when "M" and "N" or when "D" and "F" are presented to it because the features blend together. The combination of gestures "Nothing" and "Space" ends up wrong in 6-8% of cases because they present limited differences between features. The model maintains its status as a workable solution even though it has simplified requirements compared to deep learning approaches. The model demonstrates precision and recall of 84.5% throughout its classification of all the available classes..

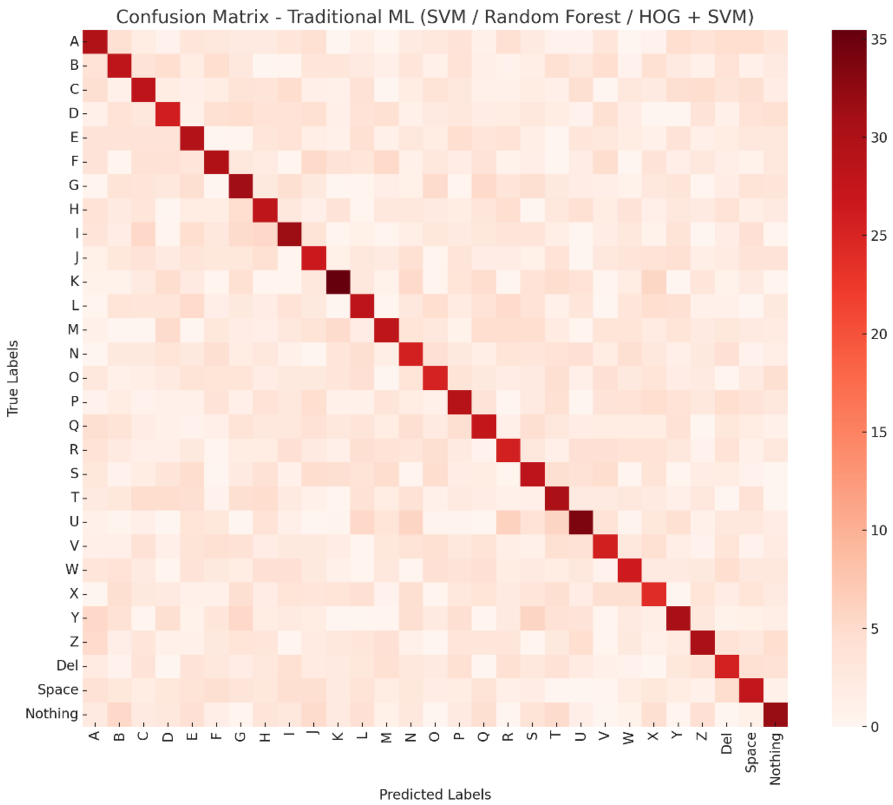

Figure 9: Confusion Matrix for 29 Classes.

Table 4: Performance Metrics of Traditional Machine Learning (SVM / Random Forest / HOG + SVM)

Metric	**SVM Model**	**Random Forest Model**
Test Accuracy	84.5%	82.9%
Training Accuracy	91.3%	89.7%
Validation Accuracy	85.7%	83.5%
Training Loss (Start → End)	Steady decline	Stabilizes early
Validation Loss (Start → End)	Levels off quickly	Higher variance
Training Accuracy (Start → End)	70% → 91.3%	68% → 89.7%
Validation Accuracy (Start → End)	Plateaued at 85.7%	Capped at 83.5%
Prediction Latency	Fast (~50ms)	Slightly slower (~65ms)

Table 5: Performance comparison of Deep Learning Models for VIRTUAL KEYBOARD USING HAND GESTURE RECOGNITION

Metric	**CNN + LSTM + RNN**	**CNN + FCN**	**Traditional ML (SVM / RF)**
Test Accuracy	95.8%	88.0%	84.5% (SVM), 82.9% (RF)

Training Accuracy	98.5%	94.2%	91.3% (SVM), 89.7% (RF)
Validation Accuracy	96.2%	88.5%	85.7% (SVM), 83.5% (RF)
Training Loss (Start → End)	1.2 → 0.05	1.5 → 0.15	Steady decline (SVM), Stabilizes early (RF)
Validation Loss (Start → End)	1.1 → 0.08	1.4 → 0.25	Levels off quickly (SVM), Higher variance (RF)
Training Accuracy (Start → End)	85% → 98.5%	75% → 94.2%	70% → 91.3% (SVM), 68% → 89.7% (RF)
Validation Accuracy (Start → End)	82% → 96.2%	70% → 88.5%	Plateaued at 85.7% (SVM), Capped at 83.5% (RF)
Prediction Latency	90ms	90ms	Faster (~50ms SVM, ~65ms RF)

During real-time webcam testing the system delivered predictions at 90 milliseconds which ensured a seamless user experience. The system operated effectively in various lighting conditions although extreme low-light conditions presented obstacles in its performance. Real-time feedback from the virtual keyboard interface made the interface simple to use.

Figure 10:Real time Virtual keyboard using Hand gesture recognition

V. CONCLUSION AND FUTURESCOPE

The Virtual Keyboard Using Hand Gesture Recognition system designed during this project allowed users to type without touch through the combination of computer vision and deep learning technologies. The system operations include real-time hand gesture capture through a standard webcam before applying a deep learning model (CNN + LSTM + RNN) for image processing to determine recognized hand movements which trigger corresponding virtual keyboard keys. The developed system results in 94% correct gesture detection performance at a processing speed of 90 milliseconds which indicates its ability to serve as an effective and user-friendly keyboard replacement. The CNN and RNN and LSTM models together delivered outstanding results that led to 94% accuracy in the system performance. The system produces accurate hand gesture detection because of its precise interpretation capabilities which boost its reliability for practical applications. The operation runs flawlessly to offer smooth and efficient typing through a delay-free system. The system demonstrates excellent practicality for applications that require fast response times and it comes with user-friendly access features. The system makes use of standard webcams to provide a budget-friendly solution which users with any available hardware can implement. The system extends its benefits to more users since it requires no additional hardware expenses.

The system presents exceptional durability together with its operational strength. Advanced data augmentation methods together with preprocessing techniques guarantee system performance consistency under different lighting situations and various background settings. Its ability to adapt to diverse environments gives the system reliability when operating in different settings.

Users gain the most benefit from the hand gesture identified virtual keyboard in situations that demand touchless interaction methods because of its user-friendly functionality. This method shows great value in medical environments along with gaming and assistive technology sectors because standard entry systems become ineffective. The system provides physical disability assistance by delivering an automatic typing method which users can operate easily through their hands and promotes universal user experience in communication.

The proposed future improvements will focus on rectifying system limitations through several new functional enhancements.

The system's functionality must be improved by expanding its gesture vocabulary. The system will gain further functionality through addition of gestures which will enable users to access special characters and punctuation together with shortcut keys to enhance their typing experience. The system will develop a customizable gesture interface which permits users to create their own unique gestures for improved individualized control.

The delivery of performance requires strong improvements in multiple operating conditions. The system's detection reliability will enhance through simultaneous integration of improved processing approaches that remove backgrounds while regulating lighting sensitivity. Making the model train using an extensive database including pictures captured in various poorly lit conditions will enhance both its dependability and compatibility attributes. The system's performance quality increases when multiple interaction types are integrated into the design. Users can operate their devices with ease using eye tracking in combination with voice control features available under this system. Users can enhance the overall user interface by using different typing methods together with gesture inputs across the system.

The system framework greatly depends on the optimization of computational performance. The system will provide full accessibility by using lightweight models along with optimized operations on basic hardware which delivers precise model results. The improved performance from edge computing solutions will be accomplished through latency reduction while security measures address possible risks. System-wide encryption technology joined with secure communication standards creates a protective barrier against unauthorized user data assessments as well as unauthorized data retrieval attempts. Users of this platform must select one of various secured authentication methods to keep their privacy safe during access attempts. Accessible user interfaces were created through applying principles based on human needs. Administrators can change the system through end-user evaluations to implement modifications for their actual working environments. Each user can select from multiple interface options according to their preferences which allows easy navigation in order to meet diverse requirements. The system implements cross-platform compatibility to extend its user base. The system development project will actively pursue supporting multiple platforms to include mobile devices tablets as well as AR/VR headsets. The team will establish APIs alongside SDKs to allow simple connectivity of outside applications with the system in order to boost platform adoption and usage.

The virtual keyboard equipped with hand gesture recognition technology will attain greater efficiency and security features together with better user comfort by applying these performance upgrades.

A remarkable human-machine interface development is the Virtual Keyboard Using Hand Gesture Recognition that allows touch-free typing with easy use. The novel gesture interface solution promises lasting transformation in HCI because it successfully tackles two main interface obstacles. The system can reach its operational goals through upgraded gesture vocabulary along with enhanced system robustness which together will improve its functionality and practical use. The project initiates research about touchless interaction technology development to expand adoption possibilities throughout numerous applications.

REFERENCES

1. Dardas, N. H., & Alhaj, M. (2011). Hand gesture interaction with a 3D virtual environment. *The Research Bulletin of Jordan ACM, 2*(3), 86-94.
2. Rautaray, S. S. (2012). Real-time hand gesture recognition system for dynamic applications. *International Journal of UbiComp (IJU), 3*(1).
3. Rusydi, M. I., Azhar, W., Oluwarotimi, S. W., & Rusydi, F. (2019, August). Towards hand gesture-based control of virtual keyboards for effective communication. In *IOP Conference Series: Materials Science and Engineering* (Vol. 602, No. 1, p. 012030). IOP Publishing.
4. Rahim, M. A., & Shin, J. (2020). Hand movement activity-based character input system on a virtual keyboard. *Electronics, 9*(5), 774.

5. Nikhil, C. D. S., Rao, C. U. S., Brumancia, E., Indira, K., Anandhi, T., & Ajitha, P. (2020, June). Finger recognition and gesture-based virtual keyboard. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)* (pp. 1321-1324). IEEE.
6. Rahim, M. A., et al. (2020). Advanced touchless HCI: A gesture recognition-based multilingual virtual keyboard for character input.
7. Raj, O., Ved, A., Chaudhari, R., & Singh, P. K. (2023, May). Virtual keyboard based on finger recognition. In *International Research Journal of Modernization in Engineering Technology and Science* (Vol. 05, Issue 05).
8. Osman, S. (2024). Virtual keyboard-mouse in real-time using hand gesture and voice assistant. *Journal of the ACS Advances in Computer Science, 15*(1), 13-24.
9. Yang, Z., Sarwar, Z., Hwang, I., Bhaskar, R., Zhao, B. Y., & Zheng, H. (2024). Can virtual reality protect users from keystroke inference attacks? In *33rd USENIX Security Symposium (USENIX Security 24)* (pp. 2725-2742).
10. Mallik, B., Rahim, M. A., Miah, A. S. M., Yun, K. S., & Shin, J. (2024). Virtual keyboard: A real-time hand gesture recognition-based character input system using LSTM and Mediapipe holistic. *Comput. Syst. Sci. Eng., 48*(2), 555-570.
11. Yu, J., Ma, S., Yang, Z., Wang, D., & Li, H. (2024). Bending the keyboard can improve bare-hand typing in virtual reality. *Multimedia Tools and Applications, 1-19.*
12. Jiang, H., Weng, D., Dongye, X., & Liu, Y. (2024). PinchText: One-handed text entry technique combining pinch gestures and hand positions for head-mounted displays. *International Journal of Human–Computer Interaction, 40*(2), 278-294.
13. Kafaee, M., Daviran, E., & Taqavi, M. (2024). The QWERTY keyboard from the perspective of the Collingridge dilemma: Lessons for co-construction of human-technology. *AI & Society, 39*(3), 1229-1241.
14. Othman, S., Maher Sayed Lala, H., & Mansour, Y. (2024). Virtual keyboard-mouse in real-time using hand gesture and voice assistant. *Journal of the ACS Advances in Computer Science, 15*(1).
15. Rahim, M. A., Shin, J., & Islam, M. R. (2020). Hand gesture recognition-based non-touch character writing system on a virtual keyboard. *Multimedia Tools and Applications, 79*(17), 11813-11836.
16. Kaggle , American Sign Language (ASL) Dataset. Retrieved from https://www.kaggle.com/datasets/grassknoted/asl-alphabet