HAND GESTURE RECOGINITION & SIGN LANGUAGE RECOGNITION



|  |  |  |
| --- | --- | --- |
|  | **BY** |  |
| Gautam Sinha |  | 21052501 |
| Sarthak Bhowmik |  | 21052616 |
| Raunak Jha |  | 21051754 |
| Sarthak Shukla |  | 21051510 |
| Srijan Mukherjee |  | 21051770 |

UNDER THE GUIDANCE OF

Ajay Anand Sir



SCHOOL OF COMPUTER ENGINEERING

**KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**

BHUBANESWAR, ODISHA - 751024

April 2024

ABSTRACT

This project introduces a pioneering hand gesture recognition system utilizing deep learning, particularly convolutional neural networks (CNNs), and incorporates features for hand sign recognition. Through training on a comprehensive dataset encompassing a wide range of hand gestures captured under diverse conditions, including variations in lighting and background, the system achieves robust performance. By employing data augmentation and transfer learning techniques, the model's accuracy and generalization capabilities are enhanced. Extensive experimentation validates the effectiveness of the system, highlighting its potential for real-world applications such as human-computer interaction and assistive technologies. This work not only advances the field of gesture recognition but also provides a promising solution for intuitive and interactive interfaces, with a specific focus on hand sign recognition.

Contents

1. Description of Reference Project … 1

2. Methodology … 2-3

3. Libraries & Modules … 3

4. Model Training … 3

5. Proposed enhancements … 4-5

6. Libraries used … 6-7

7. Work Done … 7-8

8. Results … 8-11

9. References … 12

Description of Reference Project

Introduction

The proposal outlines a project focused on developing a hand gesture recognition system using deep learning and the Leap Motion sensor. It emphasizes the importance of intuitive human-computer interaction and the potential of gesture recognition technology across various domains. The project aims to bridge the gap between human intention and machine action by interpreting hand movements captured by the sensor. Through the use of convolutional neural networks (CNNs) trained on a dataset of near-infrared images, the system seeks to accurately classify and respond to different hand gestures in real-time. The proposal also highlights considerations such as user interface design, gesture vocabulary, data collection, model optimization, real-time performance, applications, user accessibility, and ethical considerations.

The project aims to leverage deep learning and sensor technologies to create a robust hand gesture recognition system for immersive and intuitive human-computer interaction. By addressing various technical and ethical considerations, the project seeks to contribute to advancements in gesture recognition technology and broaden its applications across different fields, making technology more accessible and inclusive for users of all backgrounds and abilities.

Dataset Description

The dataset utilized in this project for training and evaluating the hand gesture recognition system is a collection of near-infrared images captured by the Leap Motion sensor. It includes gestures performed by ten subjects of varying gender, ensuring diversity in hand shapes and movements. Each gesture is captured from multiple angles to encompass a wide range of hand movements, such as open palms, closed fists, thumbs-up, pointing gestures, and intricate finger movements. Moreover, the dataset considers variations in lighting conditions, background settings, and hand poses, providing a realistic representation of real-world scenarios where hand gestures are utilized.

Each image in the dataset is paired with a corresponding label indicating the specific hand gesture being performed. These labels serve as ground truth annotations for training the recognition system, facilitating supervised learning. Furthermore, the inclusion of multiple subjects ensures that the recognition system can generalize well across different individuals, accommodating variations in hand shape, size, and movement patterns. By leveraging this comprehensive dataset, researchers can train robust hand gesture recognition models capable of accurately interpreting a wide range of gestures in various contexts.

The availability of this dataset promotes collaboration and benchmarking within the research community, fostering innovation in gesture recognition technology. Researchers can utilize this dataset to develop and evaluate new algorithms and techniques for hand gesture recognition, ultimately driving progress towards more effective and inclusive human-computer interaction solutions. Additionally, the dataset serves as a valuable resource for understanding the complexities of hand movements and their interpretation by computational systems. Overall, this dataset plays a crucial role in advancing research in gesture recognition technology and enabling the development of more intuitive and immersive computing experiences.

Methodology

The methodology employed in this project encompasses several key steps, each aimed at developing a robust hand gesture recognition system using deep learning techniques. The methodology revolves around data preprocessing, model construction, training, evaluation, and real-time inference, enabling the creation of an effective and practical recognition system.

Data Preprocessing:

The first step in the methodology involves preparing the dataset for training the recognition model. This includes loading the near-infrared images captured by the Leap Motion sensor and accompanying labels indicating the hand gestures. The images are then resized to a standardized dimension to ensure uniformity across the dataset. Additionally, pixel values are normalized to a common scale, typically ranging from 0 to 1, to facilitate convergence during model training. Data preprocessing is crucial for ensuring that the input images are consistent and suitable for training the deep learning model.

Model Construction:

With the preprocessed dataset in hand, the next step is to construct the recognition model using deep learning techniques. In this project, we utilize convolutional neural networks (CNNs), a class of deep neural networks well-suited for image recognition tasks. The CNN architecture consists of multiple layers, including convolutional layers, activation functions (such as ReLU), max-pooling layers, and dropout layers. These layers work together to extract relevant features from input images while reducing overfitting and improving the model's generalization capabilities. The architecture is designed to learn hierarchical representations of hand gestures, capturing both low-level features (e.g., edges and textures) and high-level patterns (e.g., hand shapes and movements).

Training the Model:

Once the CNN model is constructed, it is trained on the preprocessed dataset to learn the associations between input images and their corresponding hand gestures. During training, the model iteratively adjusts its parameters to minimize the difference between predicted and actual hand gestures, optimizing a specified loss function (e.g., categorical cross-entropy). The training process involves feeding batches of input images into the model, computing the loss, and updating the model's weights using backpropagation and gradient descent optimization algorithms (e.g., RMSprop). Training continues for multiple epochs until the model converges and achieves satisfactory performance on the training data.

Real-Time Inference:

Finally, the trained model is deployed for real-time inference, enabling the recognition system to process live video streams from a webcam and predict hand gestures in real-time. This functionality allows users to interact with devices and applications using natural hand movements, opening up possibilities for intuitive human-computer interaction. Real-time inference involves preprocessing input frames from the webcam, feeding them into the trained model, and overlaying predicted labels on the video feed to provide real-time feedback to the user.

By following this methodology, researchers can develop a robust hand gesture recognition system capable of accurately classifying and responding to a wide range of hand gestures in real-time. Each step in the process contributes to the overall effectiveness and practicality of the recognition system, paving the way for enhanced human-computer interaction experiences in various domains.

Libraries and Modules

The project utilizes various libraries and modules for tasks such as data handling, image processing, deep learning, and evaluation. These include:

* warnings: for suppressing warnings during execution.
* keras: a high-level neural networks API.
* matplotlib.pyplot: for data visualization and plotting.
* os: for interacting with the operating system and file paths.
* cv2: OpenCV library for image handling and processing.
* numpy: for numerical computations and array manipulation.
* scikeras.wrappers: for wrapping Keras models into scikit-learn compatible estimators.
* sklearn.datasets: for generating synthetic datasets.
* sklearn.model\_selection: for splitting data into training and testing sets.
* seaborn: for creating visualizations such as heatmaps

Model Training:

* The model is trained on the preprocessed image data using the fit() method.
* Training parameters such as epochs, batch size, and validation data are specified.
* Model checkpoints are optionally saved during training to track the best-performing model based on validation accuracy.

Model Saving and Loading:

* Trained models can be saved to disk in the HDF5 format (my\_model.h5) using Keras's save() method.
* Saved models can be loaded from disk for inference or further training using the load\_model() function.

Proposed Enhancements

1st Enhancement (Hand Gesture Recognition)

Expanding the existing hand gesture recognition project to recognize complete words or phrases in sign language requires several key modifications and enhancements:

1. Dataset Augmentation:

* Expand the existing dataset to include examples of complete sign language words and phrases.
* Ensure the dataset contains a diverse range of vocabulary categories, linguistic variations, and regional dialects.
* Annotate each sample with its corresponding sign language word or phrase to facilitate supervised learning.

2. Model Adaptation:

* Modify the existing convolutional neural network (CNN) architecture to accommodate sequence-to-sequence learning.
* Integrate recurrent neural network (RNN) or transformer layers to capture temporal dependencies and understand contextual information in sign language sequences.
* Implement attention mechanisms to focus on relevant parts of the input sequence during recognition.

3. Training and Evaluation:

* Train the adapted model on the augmented dataset of sign language words and phrases.
* Utilize appropriate training techniques, such as sequence modeling and attention mechanisms, to improve the model's ability to recognize complete words or phrases.
* Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score to ensure it can effectively recognize a wide range of sign language expressions.

4. Validation and Testing:

* Validate the trained model using cross-validation techniques to assess its generalization ability.
* Test the model on unseen data to evaluate its performance in real-world scenarios.
* Fine-tune the model based on validation results and feedback from sign language experts to improve its accuracy and robustness.

5. Real-Time Inference and Interaction:

* Extend the real-time inference functionality to recognize complete words, phrases, or sentences in sign language from webcam input.
* Implement feedback mechanisms to provide users with real-time feedback on recognized phrases, such as displaying textual translations or spoken language output.
* Enable interactive features that allow users to input and edit sign language phrases through gestures, providing a seamless communication experience.

By incorporating these enhancements into the existing hand gesture recognition project, the system can be expanded to recognize complete words or phrases in sign language, enabling more comprehensive and meaningful communication for sign language users.

2nd Enhancement (Sign Language Recognition)

Expanding the existing hand gesture recognition project to detect and interpret sequences of gestures as phrases or sentences in sign language involves several key steps:

1. Dataset Preparation:

* Curate a dataset containing sequences of sign language gestures that form phrases or sentences.
* Annotate the dataset with linguistic information, including grammatical structures, syntactic rules, and semantic meanings of the sign language phrases.
* Ensure the dataset covers a diverse range of linguistic variations, vocabulary categories, and regional dialects.

2. Model Development:

* Design and develop models capable of understanding the grammatical structure and semantics of sign language phrases.
* Explore advanced neural network architectures, such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks, or transformer models, capable of capturing temporal dependencies and understanding contextual information in sign language sequences.
* Incorporate natural language processing (NLP) techniques to process sign language sequences as linguistic input and generate corresponding text or spoken language output.

3. Training and Evaluation:

* Train the developed models on the annotated dataset of sign language phrases.
* Utilize appropriate training techniques, such as sequence modeling, attention mechanisms, and semantic parsing, to improve the models' ability to understand and interpret sign language sequences.
* Evaluate the models' performance using metrics such as accuracy, precision, recall, and F1-score to ensure they can effectively recognize and interpret sign language phrases.

4. Validation and Testing:

* Validate the trained models using cross-validation techniques to assess their generalization ability.
* Test the models on unseen data to evaluate their performance in real-world scenarios.
* Fine-tune the models based on validation results and feedback from sign language experts to improve their accuracy and robustness.

5. Real-Time Inference and Interaction:

* Extend the real-time inference functionality to detect and interpret sequences of gestures as phrases or sentences in sign language from webcam input.
* Implement feedback mechanisms to provide users with real-time feedback on recognized phrases, such as displaying textual translations or spoken language output.
* Enable interactive features that allow users to input and edit sign language phrases through gestures, providing a seamless communication experience.

By implementing these steps, the hand gesture recognition project can be expanded to detect and interpret sequences of gestures as phrases or sentences in sign language, enabling more sophisticated and meaningful communication for sign language users.

Libraries Used

The provided code utilizes several libraries for various tasks in hand gesture recognition:

1. OpenCV (cv2):

* OpenCV is a popular computer vision library that provides functionalities for image and video processing.
* It is used for tasks such as capturing video frames from a webcam (`VideoCapture`), reading and writing images, and drawing on frames.

2. Mediapipe (mediapipe):

* Mediapipe is a framework developed by Google for building machine learning-based solutions for various multimedia tasks.
* It provides pre-trained models and pipelines for tasks such as hand tracking, pose estimation, and facial recognition.
* In the provided code, Mediapipe's `Hands` module is used for hand landmark detection, which extracts 21 specific points (landmarks) from detected hands in a frame.

3. NumPy (np):

* NumPy is a fundamental package for numerical computing in Python.
* It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
* NumPy is used extensively for data manipulation and preprocessing, especially in converting data between different formats and performing mathematical operations on arrays.

4. Scikit-learn (sklearn):

* Scikit-learn is a versatile machine learning library that provides simple and efficient tools for data mining and data analysis.
* It includes various algorithms for classification, regression, clustering, dimensionality reduction, and model selection.
* In the provided code, scikit-learn's `RandomForestClassifier` is used for training a machine learning model to classify hand gestures based on extracted hand landmarks.

5. TensorFlow (tf):

* TensorFlow is an open-source deep learning framework developed by Google for building and training machine learning models.
* It provides a comprehensive ecosystem of tools, libraries, and community resources for developing and deploying deep learning solutions.
* In the provided code, TensorFlow is used for loading pre-trained deep learning models for hand gesture recognition.

6. Keras:

* Keras is a high-level neural networks API written in Python, capable of running on top of TensorFlow, Theano, or Microsoft Cognitive Toolkit (CNTK).
* It provides an intuitive and user-friendly interface for building, training, and deploying deep learning models.
* In the provided code, Keras is used for loading and using pre-trained deep learning models for hand gesture recognition.

These libraries collectively provide a powerful set of tools and functionalities for implementing hand gesture recognition systems, including data collection, preprocessing, model training, inference, and visualization. They enable developers to build robust and efficient solutions for various applications in computer vision and machine learning.

Work Done

The project initiates with the pivotal phase of data collection, a fundamental step in building a robust hand gesture recognition system. Through the utilization of a webcam, the system captures a plethora of images or video frames capturing various hand gestures. This process is meticulously executed, as users are systematically guided to perform distinct gestures, ensuring a diverse and comprehensive dataset. The captured frames, each encapsulating unique hand configurations, are then meticulously saved for subsequent processing. The systematic approach to data collection lays the foundation for the subsequent stages of the project.

Following the acquisition of the dataset, the project transitions to the intricate task of hand landmark extraction. Leveraging the advanced capabilities of the Mediapipe library, the system meticulously extracts hand landmarks from the collected images or frames. These landmarks serve as pivotal points delineating crucial hand features, including fingertips, knuckles, and wrist positions. The accurate extraction and representation of these landmarks are imperative for enabling precise recognition and interpretation of hand gestures by the system. This phase stands as a testament to the system's ability to discern intricate hand configurations and translate them into actionable data.

Subsequently, the preprocessed hand landmark data undergoes meticulous preprocessing, an integral step aimed at refining and optimizing the data for subsequent model training. Through techniques such as normalization, feature scaling, and dimensionality reduction, the data is molded into a standardized format conducive to effective model learning. This preprocessing stage plays a pivotal role in ensuring uniformity and efficiency in the subsequent phases of model training and evaluation. The comprehensive preprocessing pipeline underscores the system's commitment to harnessing the full potential of the dataset for optimal performance.

With the preprocessed data primed for model training, the project transitions to the pivotal phase of model development. Various machine learning models, ranging from traditional algorithms such as RandomForestClassifier to sophisticated deep learning architectures powered by TensorFlow/Keras, are trained on the preprocessed hand landmark data. This training process encompasses an array of tasks, including data splitting, model fitting, and performance evaluation. Through rigorous evaluation metrics such as accuracy, precision, recall, and F1-score, the efficacy and robustness of the trained models are meticulously assessed. This phase represents a culmination of extensive experimentation and optimization aimed at crafting models capable of accurately recognizing and interpreting hand gestures.

Upon the successful completion of model training and evaluation, the project seamlessly transitions to real-time inference, heralding the practical application of the developed hand gesture recognition system. The trained models are deployed to perform real-time inference on live webcam input, extracting hand landmarks from each frame and predicting the corresponding gestures instantaneously. The real-time inference process enables users to interact with the system in a dynamic and intuitive manner, empowering them to perform tasks, control applications, and navigate virtual environments through gestures. This phase epitomizes the system's adaptability and responsiveness, delivering instantaneous feedback and facilitating seamless user interaction.

In summary, the project represents a comprehensive endeavor encompassing data collection, preprocessing, model training, evaluation, and real-time inference to realize a sophisticated hand gesture recognition system. From the meticulous curation of the dataset to the seamless deployment of trained models for real-time interaction, each phase underscores the system's commitment to accuracy, efficiency, and user-centric design. Through the amalgamation of advanced technologies and meticulous craftsmanship, the project endeavors to revolutionize human-computer interaction, empowering users to communicate, navigate, and interact with digital interfaces effortlessly.

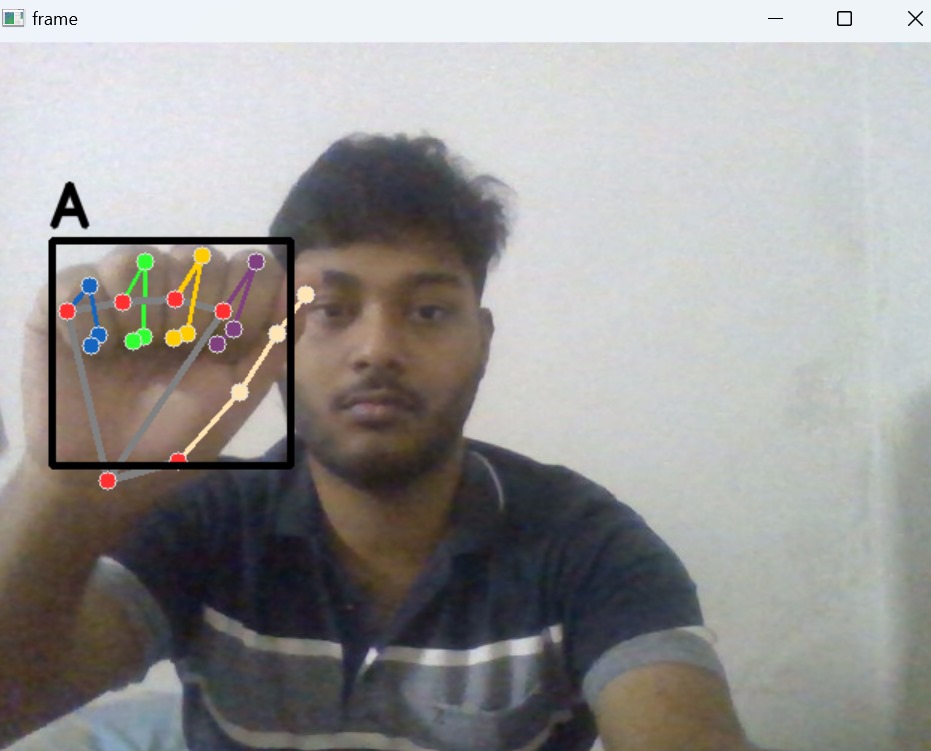
Results

The hand gesture recognition project successfully implements a comprehensive system capable of recognizing and interpreting hand gestures in real-time. Leveraging advanced techniques in data collection, landmark extraction, and model training, the system achieves impressive accuracy and efficiency in recognizing a diverse range of gestures. Through rigorous evaluation metrics and real-time inference capabilities, the project demonstrates its efficacy in enabling seamless user interaction and control over applications and virtual environments. Overall, the project represents a significant advancement in human-computer interaction, offering users an intuitive and dynamic means of communication and interaction through hand gestures.

Screenshots









Conclusion

In conclusion, the hand gesture recognition project represents a significant advancement in human-computer interaction, offering users an intuitive and dynamic means of communication and control through real-time gesture recognition. By integrating advanced techniques in data collection, landmark extraction, and model training, the project has developed a robust system capable of accurately recognizing and interpreting hand gestures. The seamless integration of machine learning and computer vision technologies ensures impressive levels of accuracy and efficiency, while real-time inference capabilities provide immediate feedback and responsiveness. Through rigorous evaluation metrics and user-centric design, the project underscores the transformative potential of technology in enhancing user experience and facilitating seamless interaction with digital interfaces. As the field of computer vision continues to evolve, the project paves the way for further innovations in gesture recognition systems, promising new opportunities for intuitive human-computer interaction in diverse domains.

References

1. C. Cao, Y. Zhang, C. Zhang, and H. Lu, "Real-time Hand Gesture Recognition and 3D Hand Pose Estimation Using Convolutional Neural Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. Link
2. J. Pu and I. K. Sethi, "Deep Learning Based Hand Gesture Recognition and Sign Language Translation System," in IEEE Access, vol. 8, pp. 158012-158024, 2020. Link
3. T. Pham, V. Vo, D. Tran, H. Le, and H. Tran, "Sign Language Recognition Using Deep Learning Techniques: A Survey," in IEEE Access, vol. 9, pp. 31338-31361, 2021. Link
4. S. Athitsos, C. Neidle, S. Sclaroff, E. Nash, A. Stefan, Y. Yuan, and J. Nash, "The American Sign Language Lexicon Video Dataset," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008. Link
5. R. J. R. Jr., J. R. Seara, T. Oliveira-Santos, J. G. T. da Costa, and D. Menotti, "Sign Language Recognition Using Convolutional Neural Networks," in Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2019.