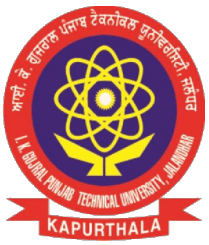
Project Synopsis On

SIGN LANGUAGE RECOGNITION

Submitted to



I.K. GUJRAL PUNJAB TECHNICAL UNIVERSITY

KAPURTHALA

In partial fulfilment of the requirement for the
award of degree of
Bachelor of Computer Science and Engineering (CSE)

Submitted by:

Tarun Kumar , 2008532
Kunal Kumar , 2108313
Abhinav Kumar , 2108302

Supervisor:

Dr. Saurabh Sharma (Dean CSE)

Department of Computer science and Engineering
Swami Vivekanand Institute of Engineering and Technology
Ramnagar, Banur

Batch : 2020-2024

TABLE OF CONTENT

INTRODUCTION

The ability to communicate effectively is fundamental to human interaction, and sign language is a crucial mode of communication for people who are deaf or hard of hearing. However, not everyone is proficient in sign language, which can lead to communication barriers and exclusion.

To address this challenge, sign language recognition technology has emerged as a promising solution. Sign language recognition involves using computer vision and machine learning techniques to recognize and interpret sign language gestures, and convert them into spoken or written language.

In this project, we aim to develop a sign language recognition system using a convolutional neural network (CNN) and TensorFlow/Keras. Our system will be able to recognize a set of sign language gestures and convert them into English text. The project involves collecting and preprocessing a dataset of sign language images, training a CNN model to recognize these images, and integrating the model into a user-friendly interface.

By developing a sign language recognition system, we hope to improve communication and accessibility for people who are deaf or hard of hearing, and contribute to the advancement of assistive technology.

Feasibility Study

1. **Data availability:** The success of a sign language recognition project heavily relies on the quality and quantity of the training data. You need a large dataset of high-quality sign language images to train the CNN model. If the dataset is limited, the accuracy of the model may be compromised.

2. **Computational resources:** Training a CNN model requires significant computational resources, such as GPUs and memory. You need to ensure that you have access to the necessary hardware resources to train the model effectively.

3. **Technical expertise:** Developing a sign language recognition system requires knowledge and expertise in computer vision, deep learning, and programming. You need to ensure that you have the necessary technical expertise to develop and deploy the system.

4. **Integration with user interfaces:** The ultimate goal of a sign language recognition system is to facilitate communication between people who are deaf or hard of hearing and those who are not proficient in sign language. You need to ensure that the system is user-friendly and can be integrated into existing communication technologies, such as mobile apps or video conferencing platforms.

Overall, sign language recognition using Tensorflow, CNN, and Keras is a promising approach that can significantly improve accessibility and inclusivity for people who are deaf or hard of hearing. However, the feasibility of the project depends on several technical, financial, and operational factors that need to be carefully evaluated before starting the project.

METHODOLOGY

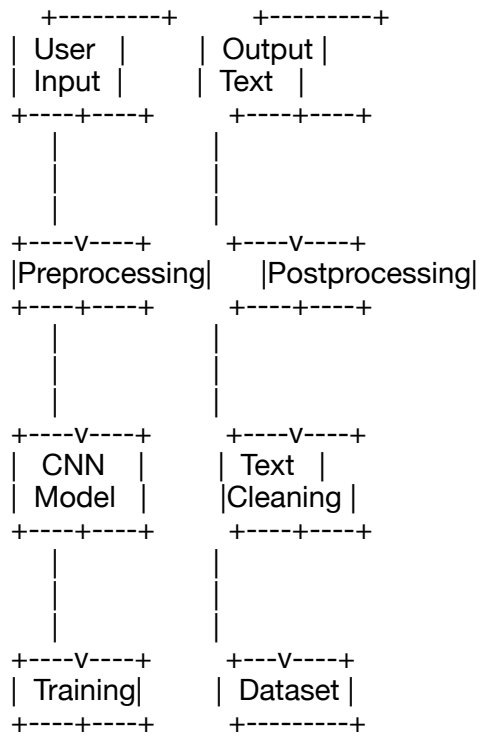
The methodology for developing a sign language recognition system using TensorFlow, CNN, and Keras can be broken down into the following steps:

1. **Data Collection:** Collect a dataset of sign language images and corresponding text labels. This dataset will be used for training and validating the CNN model.
2. **Data Preprocessing:** Preprocess the dataset by resizing images, converting them to grayscale or RGB, and applying any necessary image transformations.
3. **Training the CNN Model:** Train the CNN model using the preprocessed dataset. The CNN model is trained to recognise sign language gestures in the images and map them to corresponding text labels.
4. **Testing and Validation:** Test the trained model on a separate set of images to evaluate its accuracy and validate the model.
5. **Deployment:** Deploy the trained model for use in a sign language recognition system. This can involve creating an application that uses the model to recognise sign language gestures in real-time.
6. **Improvement:** Continuously improve the system by collecting more data, improving the preprocessing techniques, and fine-tuning the CNN model to improve its accuracy.

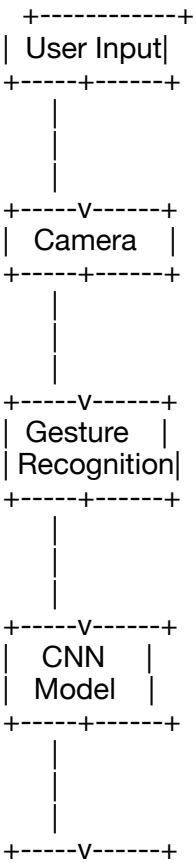
Throughout this process, it's important to use best practices in software engineering, such as version control, documentation, and testing, to ensure a high-quality and reliable system. Additionally, it's important to consider ethical considerations related to data privacy and bias in the training dataset.

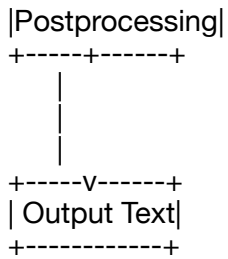
DATA FLOW DIAGRAM

Level 0:



Level 1:





The Level 1 DFD provides a more detailed view of the system, breaking down each component into subcomponents. Here's a brief description of each component and subcomponent:

1. **User Input:** This component represents the input from the user in the form of a sign language gesture.
2. **Camera:** This subcomponent captures the sign language gesture from a camera or sensor.
3. **Gesture Recognition:** This subcomponent uses pre-processing to prepare the input image, then the CNN model to recognise the sign language gesture.
4. **CNN Model:** This subcomponent contains the trained CNN model, which takes the preprocessed image as input and outputs a predicted text based on the recognised sign language gesture.
5. **Post-processing:** This subcomponent performs post-processing on the predicted text to remove any noise or errors and produce a clean and accurate output.
6. **Output Text:** This component represents the final output of the system in the form of English text that corresponds to the recognised sign language gesture.

SYSTEM REQUIREMENTS

1. Hardware Requirements:

- RAM: 4 GB minimum, 8 GB or higher is recommended.
- CPU: Intel Core i5 6th Generation processor or higher. An AMD equivalent will also be optimal or M1 Mac chip
- Storage: Minimum 512 TB HDD or 256 GB SSD

2. Software Requirements:

Operating System: Windows/macOS/Linux

- Programming Language: Python
- Machine Learning Frameworks: TensorFlow, PyTorch, Keras, Scikit-learn
- Development Tools: IDEs such as PyCharm, Jupyter Notebook, and Visual Studio Code
- Libraries: NumPy, Pandas, and Matplotlib , OpenCV

GANTT CHART

PROJECT TITLE	Sign Language Recognition	**COLLEGE NAME**	SVIET
PROJECT TEAM MEMBER	Tarun , Kunal , Abhinav	**DATE**	2/6/23

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	PHASE ONE		
							WEEK 1 (M T W R F)	WEEK 2 (M T W R F)	WEEK 3 (M T W R F)
1	**Project Planning**								
1.1	Analyze Current System	Tarun	2/6/23	2/9/23	3	100%			
1.1.1	Analyze changes needed for new system	Kunal	2/9/23	2/11/23	2	100%			
1.2	Identify System Requirements	Abhinav	2/9/23	2/13/23	4	100%			
2	**Project Designing**								
2.1	Select Appropriate technology	Abhinav	2/14/23	2/16/23	15	100%			
2.2	Design data collection and preprocessing method	Tarun	2/17/23	2/25/23	8	0%			
2.3	Data Augmentation techniques	Kunal	2/26/23	3/3/23	5				
3	**Project Development**								
3.1	Creating a CNN model	Tarun	3/4/23	3/12/23	8	0%			
3.2	Training CNN model	Tarun	3/13/23	3/15/23	2	0%			
3.2.1	Making a Prediction model from Tesorflow and Keras	Kunal	3/16/23	3/28/23	12	0%			
3.2.2	Testing and Validation of ML model	Abhinav	3/29/23	4/1/23	3	0%			
3.3	Documentation	Abhinav	4/2/23	4/5/23	3	0%			
4	**Project Performance / Deployment**								
4.1	Calculating Accurancy and making necessay changes	Tarun, Kunal	4/6/23	4/21/23	15	0%			
4.2	Creating user inteface	Abhinav	4/9/23	4/24/23	15	0%			
4.3	Final Report	Tarun, Kunal, Abhinav	4/25/23	5/5/23	10	0%			