A Mobile Application for Nepali Sign Language Detection Using Deep Learning

Subhash Belbase

Master's Thesis

MEng in Big Data Analytics

2024

Degree Thesis

Subhash Belbase

A Mobile Application for Nepali Sign Language Detection Using Deep Learning.

Arcada University of Applied Sciences: MEng in Big Data Analytics, 2024.

Commissioned by:

No Commissioned

Abstract:

Sign language greatly benefits hearing-impaired individuals by enabling effective communication with the world. However, understanding and decoding sign language can be challenging for those unfamiliar with it. To bridge this gap, we propose developing a Nepali Sign Detection application using Flutter and leveraging ResNet-50 and VGG-16 deep learning algorithms. ResNet-50 outperforms VGG-16 with a higher F1 score (0.9333 vs. 0.8475), indicating superior precision, recall, and balance in detecting positive cases.

To simulate actual network circumstances, we purposefully include delays in the creation of our mobile applications. It incorporates network inquiries using Flutter, camera access, and easier permissions management with ease. In order to mimic network latency, images are purposefully transferred to the Flask backend with pauses. To load, anticipate, and resize images, backend technologies like Pillow, Torch, and torchvision are utilized. To achieve accurate processing simulations, deliberate pauses are incorporated during model loading. The application's real-world robustness is increased by this purposeful delay integration, which guarantees authentic information flow.

This application objective is to support both Nepali hearing impaired community and the people who are strange to Nepali Sign Language (NSL) in that they can recognize and understand NSL sign direct in real time.

Keywords: Machine Learning, Deep Learning,VGG16, ResNet-50, Gesture recognition, Nepali sign language, Mobile application, Flutter framework

Contents

Figures

Tables

Abbreviations

ML	Machine Learning
NSL	Nepalese Sign Language
NPL	Natural Processing Language
SLR	Sign Language Recognition
CNN	Convolution Neural Network
D&D	Dumb and Deaf
FISPAN	Flutter Integration for Seamless PANel Management
HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
API	Application Programming Interface
CV	Computer Vision
ANN	Articifial Neural Network
ASL	American Sign Language
ASL	Australian Sign Language
BSL	British Sign Language
DSL	Danish Sign Language
FSL	French Sign Language
US	United States
UK	United Kingdom
HSLR	Hand Sign Language Recognition
AR	Augmented Reality
CVPR	Computer Vision and Pattern Recognition

Forward

I made this master's thesis to be all about research. The goal was to ensure it's based on solid research methods. I would like to express my sincere gratitude to everyone involved with the Arcada University of Applied Sciences Master of Big Data Analytics program, particularly my supervisor, Leonardo Espinosa-Leal, as well as Magnus Westerlund and Truong An Pham. Their invaluable feedback greatly contributed to the completion of this research. I am grateful for you sharing the data, Jatin Bhusal, Dipen Boyaju, Nir Ratna Shakya, and Sonia Dhaubhadel.

I've thoroughly enjoyed this course; it's been a truly enriching experience for me. The content has been engaging, and I've found myself eagerly diving into each topic. From the insightful discussions to the practical exercises, every aspect of the course has contributed to my learning journey in a meaningful way. Overall, I'm grateful for the opportunity to be part of such a rewarding learning experience.

1 Introduction

1.1 Background

According to the World Health Organization's (WHO) report from February 2, 2024 (Deafness and hearing loss, 2024), by the year 2050, it's predicted that about 2.5 billion people worldwide will have some level of hearing loss. Among them, at least 700 million individuals will need assistance through hearing rehabilitation services. Shockingly, over 1 billion young adults are at risk of permanent hearing loss due to unsafe listening practices. The report suggests that a small yearly investment of less than US$ 1.40 per person could significantly expand ear and hearing care services globally. This investment could greatly improve the quality of life for millions of people affected by hearing impairments. The primary communication method used by the Deaf and hard of hearing people is sign language; it is a visual language system supported by non-verbal symbols which are created using specific sign pattern. Various countries can invent their own sign language that differs from the lexical and syntactic constructions that are native to the language. For example, For deaf communities, a variety of SLs including Nepali Sign Language (NSL), American Sign Language (ASL), Australian Sign Language (ASL), British Sign Language (BSL), Danish Sign Language (DSL), French Sign Language (FSL), and many others have been established. NSL is the sign language that is extensively used by the Deaf and Dumb (D&D) community in Nepal, and an uncommon problem for communicating and interpretation of all happenings arises in such a case.

According to the National Federation of the Disabled - Nepal (NFDN), data from the Nepal Census of 2078 reveals that out of a total population of 29,164,578, approximately 2.2% have some form of disability. Among males, the percentage is slightly higher at 2.5%, while among females, it's slightly lower at 2.0%. The census further highlights specific disability types: 7.85% of the population is deaf, 7.87% are hard of hearing, and 6.36% have speech impairments. In terms of actual numbers, this translates to 51,373 deaf individuals, 51,520 hard of hearing individuals, and 41,676 people with speech impairments in Nepal (Admin, 2023). Though NSL is of utmost significance at present, there have been a smaller number of research studies and development works and interpretation and translation of the natural language to a greater extent (Ligal & Baral, 2022).

For the purpose of the Nepali Sign Language (NSL), below Figure 1 shows numerals and consonants in Nepali Devanagari. With its clear depiction of the alphabet and number

symbols utilized in the context of Nepali Sign Language, this visual aid probably acts as a reference for NSL users or learners.

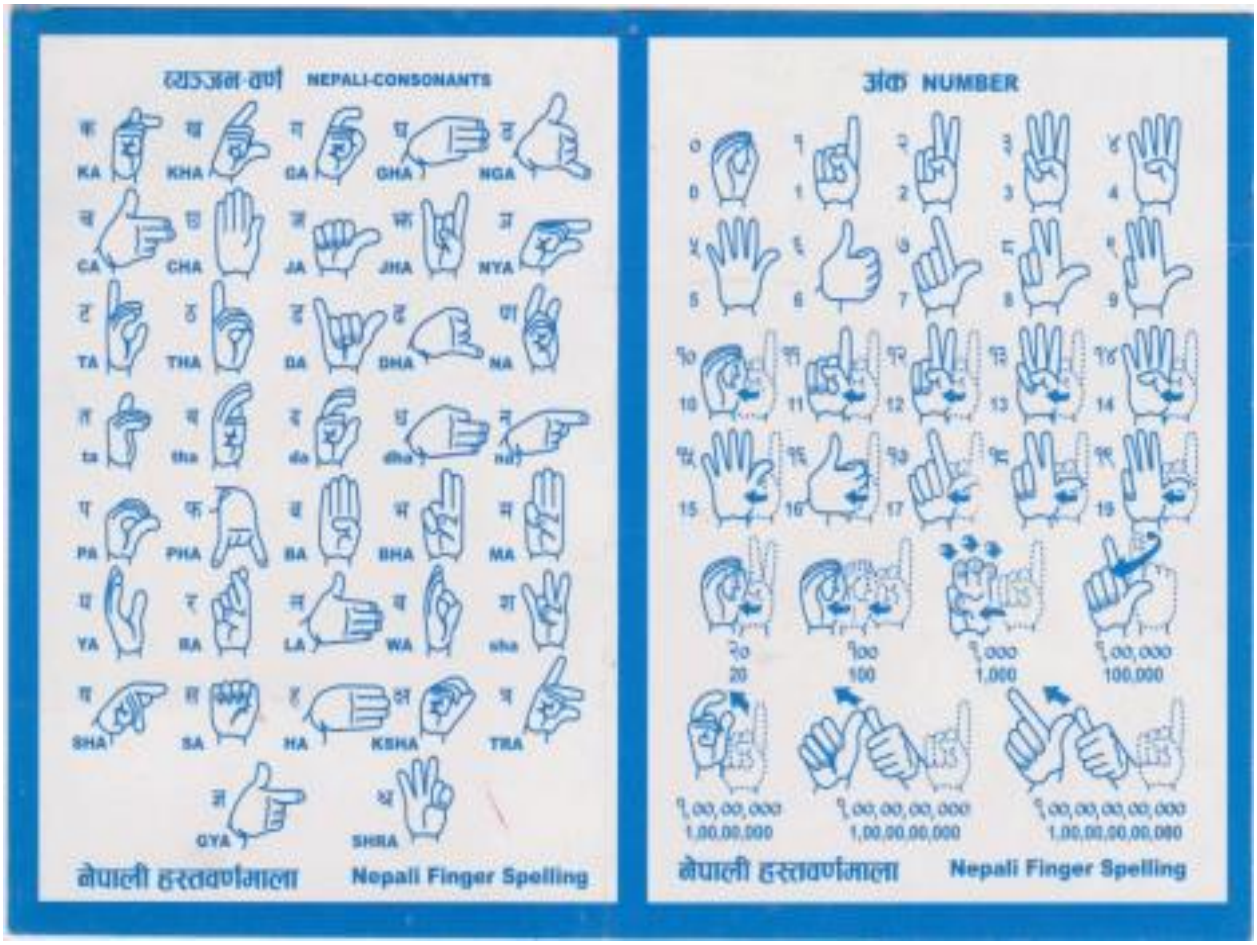


Figure 1.Nepali Devanagari Consonants and Numbers for NSL (Gurung, 2017).

1.2 Statement of Problem

Interpreting Nepali Sign Language (NSL) poses significant challenges due to several factors, including the reliance on unreliable datasets, insufficient research efforts, and the lack of connectivity between Deaf and Dumb (D&D) individuals and the wider community. These obstacles highlight the urgent need for a creative approach aimed at enhancing comprehension and accessibility of NSL.

1.3 Motivation

A significant gap that has been identified between the generally story informed D&D individuals and the generally uninformed the general populace in NSL energizes FISPAN project development. Furthermore, the general goal of this project is to provide a dataset

which is vital for more research in NSL; thereby giving a base for future research endeavors and inspiring students and researchers out there in the field.

1.4 Objective

The main objectives of the research includes:

- Model Development: To create an accurate machine learning model capable of recognizing static and dynamic sign gestures in Nepali Standard Sign Language.
- Build a System for Sign Language Translation: To construct a system that translates detected sign gestures into corresponding Nepali text, aiding communication for hearing-impaired individuals.
- Design an Mobile Application for Practical Use: To develop an user-friendly Mobile application interface where individuals can interactively use the sign language recognition system in real-time.

1.5 Scope and Application

The proposed model and application aim at facilitating the communication with the D&D public who have the Nepali alphabets by effecting a device application that will be able to detect specific Nepali alphabets and numbers also. This system fulfills these criteria permitting online D&D enthusiasts to enjoy in various ways.

1.6 Limitation

As of now, the designed system and application can only achieve a partial set of facial gestures as well as very few Classes of Nepali commonly used static and dynamic Sign gestures. Adding to this limitation prompts for the development of the large datasets in training and designing different algorithms. Following up will be the upgrades of those parts with the addition of gesture recognition and the improvement of the overall system more efficiency.

2 Methods Review

2.1 Overview

To make the things right, we look through all data we have at our disposal. The Vision-based approach has been included in team employments to create an Agile Development Method to design the project development process Thus, since the NSL is more of the manual gesture's signs rather than devices for interaction are not necessary. The project methodology is divided into four main sections:

a) Image Capturing and Preprocessing,
b) Model Generation & Training,
c) Prediction,
d) Platform Deployment.

Each section of the solution carries out the best functioning from the database of the captured and reprocessed images to training deep learning models, predictions, organizing the solution, and finally deployment on the respective platforms. Such systematic way guarantees the whole process of interpreting NSL, resulting in the highest efficiency and effectiveness to face with address the problem (Rum & Boilis, 2021).

2.2 Requirement Analysis

The used in this phase where technical specification identify, and document accumulated which were all meticulous. To get the start of large-scale research papers, trending in the field was explored by studying different available sources and relevant literature. Ultimately, the collected information was arranged following the grammatical scheme of Nepali so that a holistic view and understanding of the language as it exists in the country is achieved. The systematic method pursued was so profound that it enabled the realization of perfect finding of the design specification, which in turn directed conception and development of the system. Bringing forth nuances belonging to the existing literature and sign linguistic features has enabled the development team in follow up efforts that close all the gaps between the system functions and these unique features of Nepali sign language interpretation (Mali, Mali, Sipali, & Panday, 2018).

2.3 Related Work

Previous studies have looked into a variety of accessibility and sign language interpreting issues in various nations. Research has looked at the creation of deep learning-based systems for the recognition of sign language in the US, UK, and Japan, among other nations. Furthermore, studies have concentrated on the difficulties Deaf and Dumb (D&D) people encounter while trying to get healthcare, education, and work prospects in a variety of cultural contexts, such as South Africa, Australia, and Canada. Moreover, nations such as Sweden, India, and Brazil have been working on developing mobile applications and technical solutions for communication assistance and sign language translation. The global landscape of sign language research is enriched by these studies, which also emphasize the significance of removing cultural and linguistic obstacles to advance inclusion and accessibility for people with disabilities around the globe.

The Nepali language sign interpretation is one area of research which through NSL interpretation has played vital role of sign language. These studies have all been dedicated to a situation in which the aforesaid group of researchers developed a recognition system for NSL. The manner research employed deep learning programs to highlight several NSL hand gestures was highly precise, thereby improving communication ability for the deaf community.

Also, NSL23 dataset paper discusses the creation of a dataset for Nepali Sign Language (NSL) alphabets, aiming to facilitate the learning and development of technologies for NSL translation. The dataset, NSL23, comprises videos demonstrating NSL alphabet signs performed by volunteers under various conditions, including different lighting and environmental setups. The dataset includes gestures for vowels and consonants, totaling 1205 gestures performed by 14 volunteers, classified into beginners and experts in NSL usage. NSL23 fills a gap in research by providing a comprehensive dataset for NSL, enabling the training of machine learning models for alphabet classification and the development of sign language translation systems (Sunuwar, Borah, & Kharga, NSL23 dataset for alphabets of Nepali sign, 2024).

”Finger Spelling Gesture Recognition for Nepali Sign Language Using Hybrid Classical Quantum Deep Learning Model” by Dipen Boyaju, Jatin Bhusal, Nir Ratna Shakya, Sonia Dhaubhadel project enable direct communication between people who are deaf and people who are not familiar with sign language by developing a sign language recognition system specifically for Nepali Sign Language (NSL) gestures. Neural networks are used to recognize

sign language because they are more accurate and straightforward than traditional methods, which have certain limits. In order to recognize and classify six different NSL gestures, the project presents a hybrid classical-quantum deep learning model. It extracts visual features using the VGG16 architecture, transforms them, and then embeds them into a Variational Quantum Circuit (VQC). Better performance is the outcome of combining classical and quantum neural networks, which shows promise for the development of more practical and accessible communication solutions for the deaf and mute community (Boyaju, Bhusal, Shakya, & Dhaubhadel, 2023).

Moreover, Sanyukta Ligal and his colleagues contributed to this field by studying the challenges/opportunities of using machine learning approaches relative to NSL interpretation. Their study concluded on data selection and preprocessing technique that are the essential factors to enhance the reliability and resilience of known speech systems. As well as this, team has examined elaborating a mobile chatting application that would be running on gestures-into-text or audio instead of interpretation. Their objective was to improve the awareness about communication between NSL consumers and non-signers, so that those signing their mother tongue would become visible and accessible in multiple surroundings (Ligal & Baral, 2022).

Recent study by (Asif, Shrikhande, Pingale, Joshi, & Sonawane, 2024) in 'Hand Sign Language Recognition Using Augmented Reality & Machine Learning' highlights the importance of utilizing technology breakthroughs to empower people with various linguistic needs and close communication gaps in order to address this dilemma. The HSLR app is a game-changer in addressing this issue by empowering IWSHI to speak with confidence. Their software uses cutting-edge technologies like Augmented Reality (AR) and Machine Learning (ML) to recognize hand signs in real time and translate them into English instantly, facilitating smooth conversation. Incorporating augmented reality technology also improves the user experience by providing interactive and engaging sign-language communication platforms. Because of the large dataset we provided, the MediaPipe model deployed in real-time delivers great accuracy in sign language recognition.

2.4 System Design

In this system design section, we'll discuss how our system operates. First, let's look at a "system diagram," which illustrates all the various components of our system and how they

fit together, much like a puzzle. This gives us an overall view before we explore each part in detail to understand how they work together smoothly.

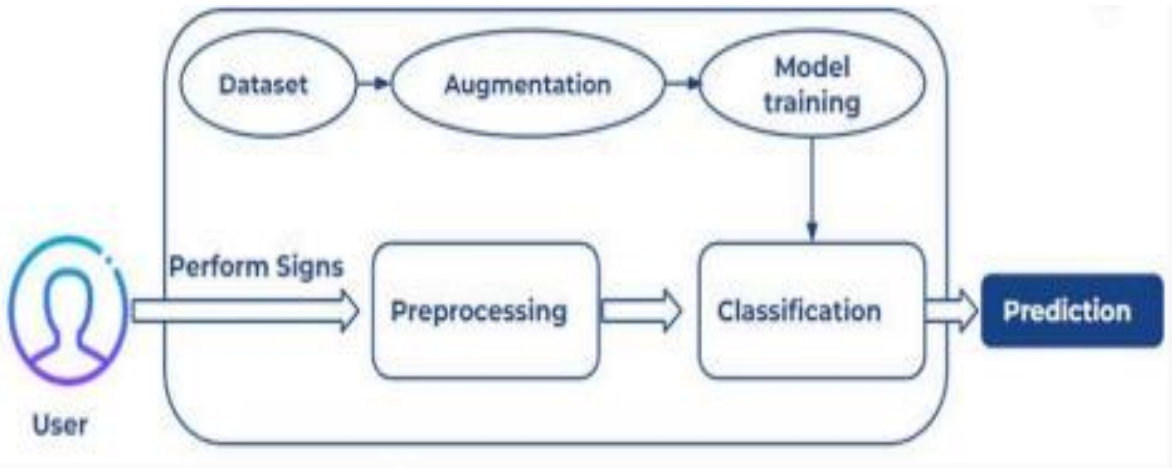
The system diagram Figure 2 illustrates the overall architecture of the system as well as its component pieces. It also shows how different modules interact with one another and support the system's functionality.

An overview of user interactions and system behavior is given by the general usage flowchart Figure 3, which shows the steps or procedures that must be followed in order to utilize the system.

Together with the actors or users engaged in each use case, the use case diagram Figure 4 shows the different activities or features that users may do within the system.

The entity-relationship (ER) diagram Figure 5, which displays the relationships between different entities or data components in the system, provides an explanation of how data is arranged and connected.

The context diagram Figure 6 provides a more thorough understanding of the system in connection to its surroundings with regard to the external entities or systems that interact with the system and the nature of these interactions.



.

Figure 2. System Diagram

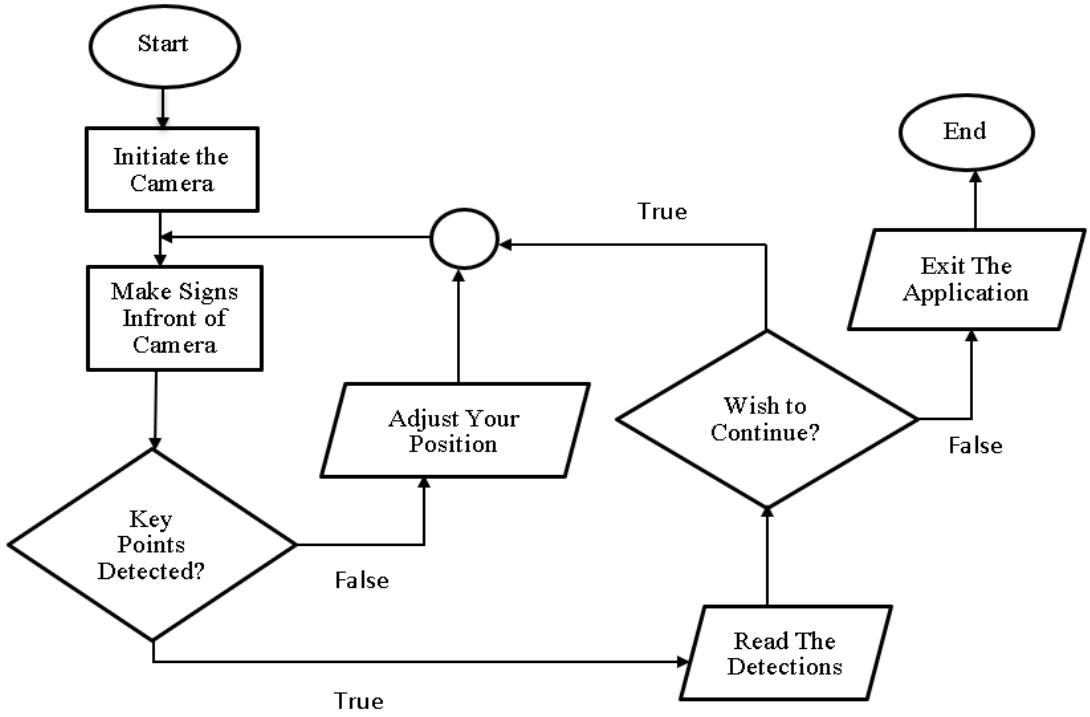


Figure 3.General Usage Flowchart

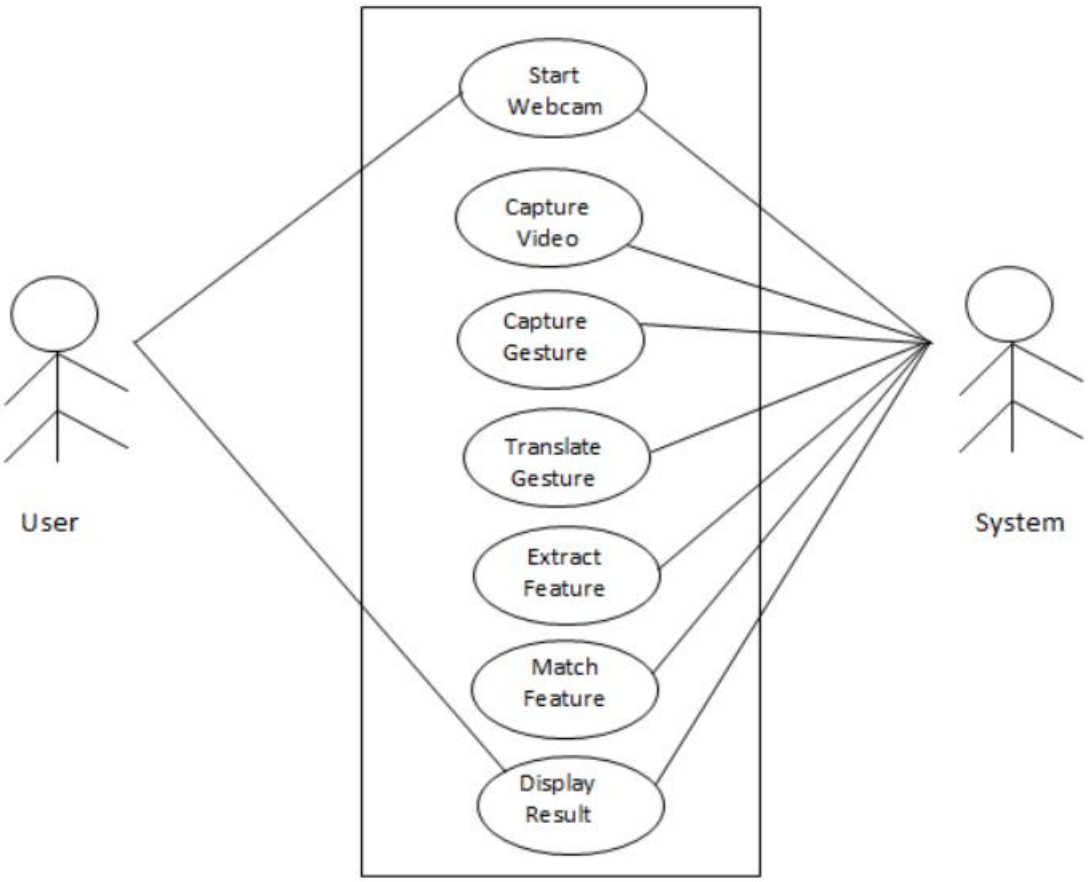


Figure 4.Use Case Diagram

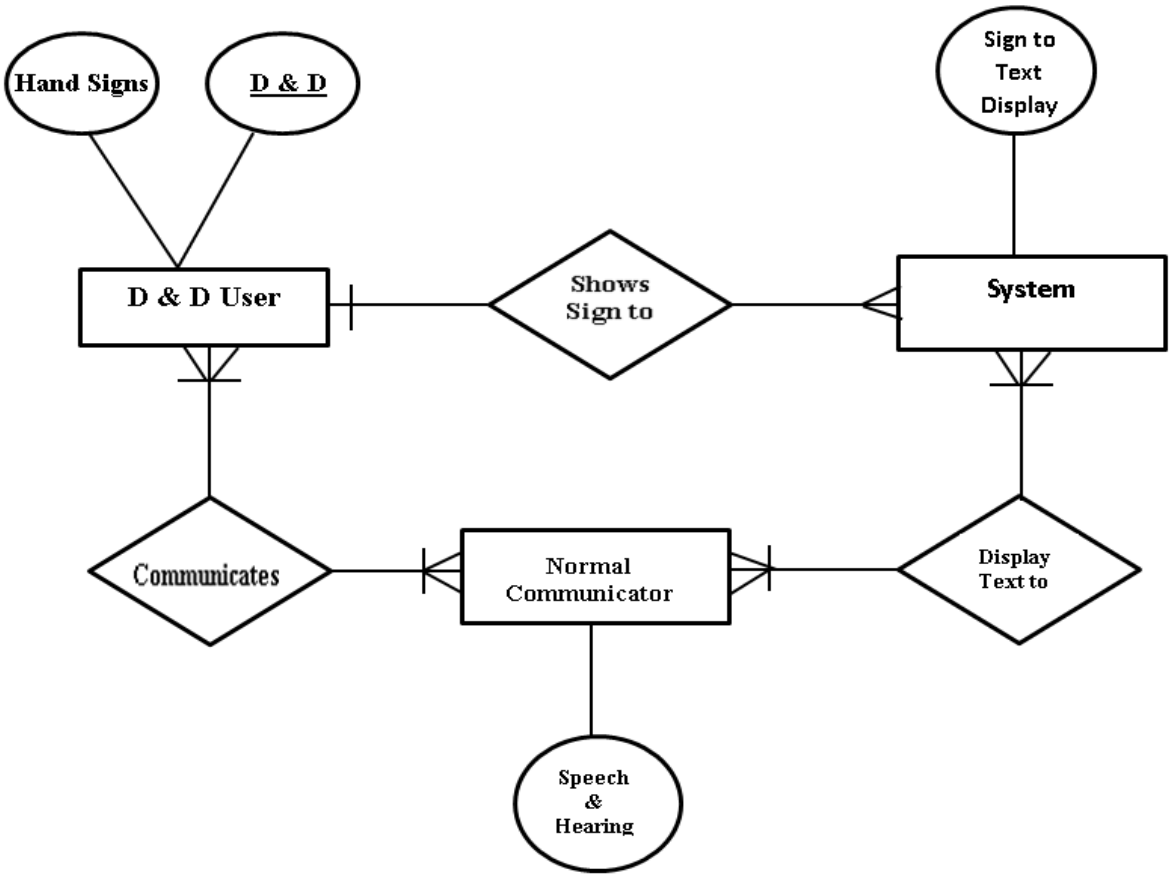


Figure 5.ER Diagram

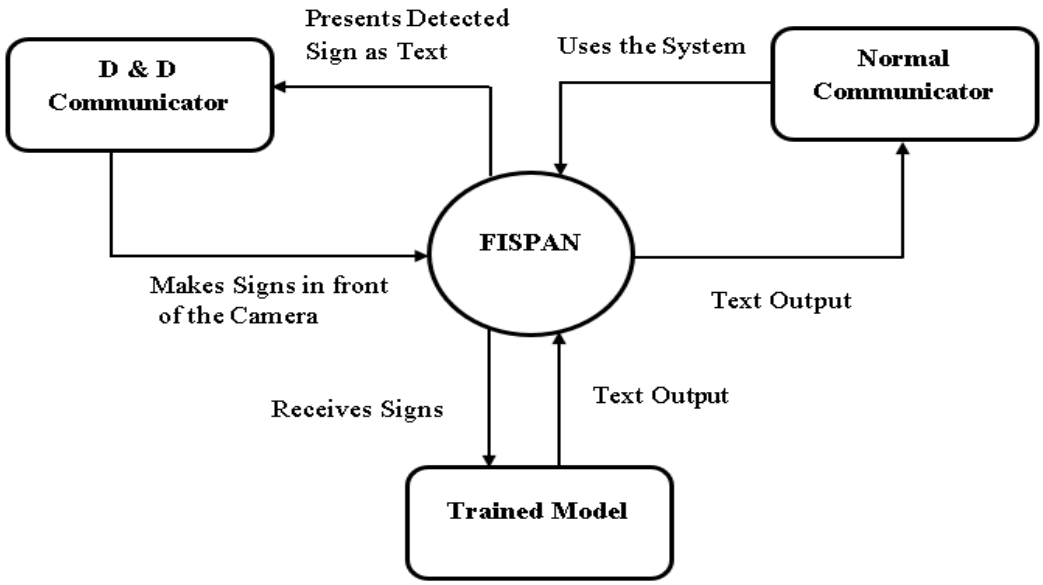


Figure 6.Context Diagram

2.5 Experimental Setup

We used two platforms for the experimental setup to satisfy the different needs of developing mobile applications and training our models.

Firstly, the model was developed in a Jupyter Notebook environment using Pytorch and Python. The model's parallel processing capabilities were powered by Nvidia GPUs, specifically the 'Cuda' architecture. The configuration details are as follows: Python version 3.11.3, Cuda version 11.8, Pytorch version 20.1, NVidia GeForce RTX 2050TI GPU with 12GB RAM, running on Microsoft Windows 11.

Secondly, for mobile application development, we picked for a core Flutter library, incorporating features such as camera access, network requests using the http library, seamless permissions management, and image conversion. The application captures images at regular intervals using the device's camera and sends them to the backend server via HTTP REST API requests. Backend operations are managed by Flask, with Torch utilized for model predictions, torch vision for tasks like resizing, and Pillow for image loading. The Torch library loads the model, enabling an endpoint '/predict' to receive, preprocess, generate questions, and respond with predictions in JSON format. This seamless integration between the Flutter frontend and Flask backend ensures smooth information flow from image capture to prediction display. Additionally, Java and Visual Studio were utilized for development purposes.

2.6 Data collection

2.6.1 Dataset

The dataset relied in this research has classes 4 with Nepali Sign Language (NSL) static gestures in atomism like “Namaskaar”, “Ghar”, “Dhanyabad” & “Ma”. We are grateful for the free dataset that Jatin Bhusal and the happy Kaggle team from Nepal provided as well. Dataset's composition helps this process with the training and evaluation of that proposed model following all the gestures of Non-Signed Language gesture (Boyaju, Bhusal, Shakya, & Dhaubhadel, 2023).

2.6.2 Dataset Size

To making the model more accurate and capable to generalize, several samples for each class of the dataset are collected, in sufficient number. This way, the model is profited from the

abundance of the dataset samples whereby the model can learn perfectly from each specific gesture and in turn improving its recognition performance.

2.6.3 Data Collection for Data Preparation Training

The dataset relied in this research has classes 4 with Nepali Sign Language (NSL) static gestures in atomism like “Namaskaar”, “Ghar”, “Dhanyabad” & “Ma”. We are grateful for the free dataset that Jatin Bhusal and the happy Kaggle team from Nepal provided as well. Dataset's composition helps this process with the training and evaluation of that proposed model following all the gestures of Non-Signed Language gesture.

2.6.4 Expected Outcomes

Static and Dynamic Gesture Recognition: The identification of the main result of such research will be a reliable model for imitating static and dynamic NSL gestures. A lifelong bid to realize the high level of accuracy and reliability is set be possible with the machine learning methods leveraging the data used and implemented.

Improved Accessibility: The NSL language recognition model can be used as the basis of a variety of applications in the accessibility for the population of hard of hearing and deaf people. It has a reasonable potential to improve the quality of life of underserved individuals. Through sign language interpretation assisted communication, the model allows for comprehensive involvement of disabled of hearing in different spheres of life.

Validation and Generalization: By giving the model trial and test to establish its capability on prediction and performance, a model validation and generalization will be assessed. It is anticipated that the model will successfully tackle the challenge of correct classification accuracy on a variety of NSL signs and demonstrate consistency on unseen cases showing thereby its applicability in real-world applications.

Practical Implications and Academic Contribution: The applied proposed model can have an academic, information and accessibility impact in different areas such as education, communication, and jobs. Also, this study can be used by researchers in the field of non-native sign language interpretation as well as use of ML in SLR. Moreover, it gives the means and methodologies which can be used in future research in the field.

2.7 Transfer Learning

We utilize a multi-step procedure in our transfer learning technique using the ResNet-50 model to modify the pre-trained model to our particular purpose. First, we load the already trained ResNet-50 model, using the vast amount of knowledge it has learned while training on the ImageNet dataset. With its extensive collection of feature representations that may be tailored to our intended job, this model is a potent beginning point. After loading the model, we apply a parameter freezing technique that, in essence, "locks" all of the pre-trained model's parameters to prevent it from receiving updates during training. As a result, key features are retained during the adaptation process, and the weights and biases of the model retain the learnt knowledge.

Now that the basic components have been developed, we modify the ResNet-50 architecture to meet the requirements of our specific mission. To account for the exact number of classes in our dataset (four in this example), we have to modify the last fully connected layer of the model. Customizing the output layer to match the unique properties of our dataset and alter the model's predictions enables more accurate classification results. Additionally, the model's architecture includes a LogSoftmax layer that effectively normalizes the output probabilities, enhancing the model's predictability and interpretability.

To ensure efficient use of computer resources, we carefully move the changed model to the chosen device—a GPU or CPU—based on availability and performance parameters. By exploiting hardware acceleration characteristics, such GPU acceleration, we can expedite the deployment process and give faster inference speeds during training. This meticulous attention to hardware optimization improves the transfer learning pipeline's overall efficiency and scalability, making it simple to integrate into real-world applications.

In summary, we leverage the strengths of the ResNet-50 model in our transfer learning technique while customizing it to the unique characteristics of our target issue. By using a structured approach that combines hardware optimization, architectural change, and parameter freezing, we improve the effectiveness of the transfer learning process. As a result, a customized model with exceptional accuracy, efficiency, and flexibility for a variety of applications is produced.

2.7.1 VGG-16

There are 16 weights in the VGG-Net. There are three completely connected layers and thirteen convolutional layers in this arrangement. After a convolutional layer, a ReLU activation function is applied, and at each level, the number of max-pooling layers is

increased to the corresponding spatial dimensions. The network is based on the use of 3x3 convolutional kernels at every layer. This promotes going deeper while keeping the number of parameters to a minimum. The inception module repeats the final layers which are fully connected. There are no more layers than the last one, a softmax classifier for the appropriate number of classes in the dataset. VGG-16 is recognized for its simplicity and evenness in architecture, giving it the advantage of being comprehend easily and achieves near to the competitive performance on image data classification tasks.

VGG16, also referred to as VGGNet, is one of the most widely used CNNs for image classification applications. With 16 layers, it is the convolution neural network (CNN). It is capable of supporting sixteen layers.. In their paper, very deep convolutional networks for large scale image recognition (VirOn), guys at Oxford University (Simonyan & Zisserman, 2015) suggested this model.

The VGG-16 Architecture, a well-known deep convolutional neural network, is shown in Figure 7. Its structure and functions are shown by seeing the convolutional and pooling layers in a sequential order.

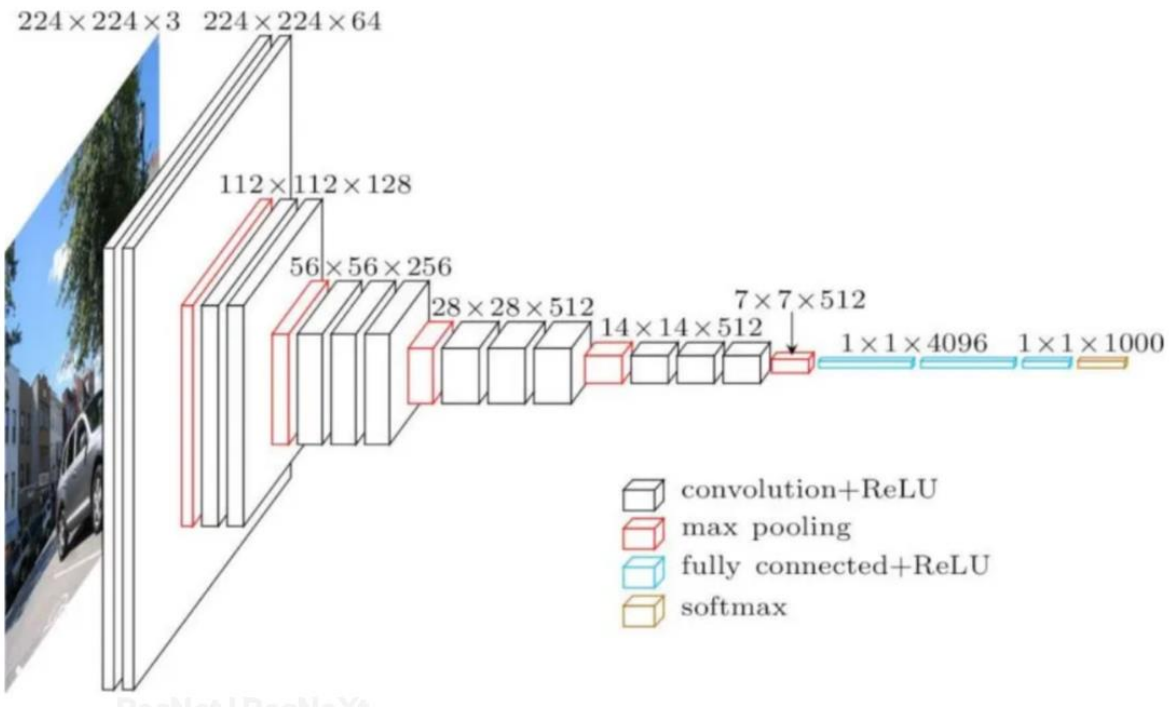


Figure 7.VGG-16 Architecture((Simonyan & Zisserman, 2015)

The VGG-16 architecture, developed by Simonyan and Zisserman in 2015, is a convolutional neural network (CNN) primarily used for image classification tasks. Here's how it works:

Input Layer: The network takes an input image of fixed size (typically 224x224 pixels) as its input.

Convolutional Layers: VGG-16 consists of 16 convolutional layers, with each layer performing feature extraction by convolving input images with a set of learnable filters (also known as kernels). These filters detect patterns and features at different spatial scales and levels of abstraction.

ReLU Activation: After each convolutional layer, a rectified linear unit (ReLU) activation function is applied elementwise to introduce non-linearity into the network, enabling it to learn more complex patterns and features.

Max Pooling Layers: In between convolutional layers, max pooling layers are used to downsample the feature maps, reducing their spatial dimensions while retaining the most important information.

Fully Connected Layers: Towards the end of the network, there are several fully connected layers followed by ReLU activations. These layers consolidate the extracted features from earlier layers and perform classification based on these features.

Softmax Layer: The final layer of the network is a softmax layer, which assigns probabilities to each class based on the features extracted by the previous layers. The class with the highest probability is considered the predicted class for the input image.

2.7.2 ResNet

ResNet-50 is a deep convolutional neural network architecture proposed by (He, Xiangyu , Shaoqing, & Jian , 2016) in their seminal paper titled "Deep Residual Learning for Image Recognition" published in the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2016. It is a member of the ResNet architectural family, which is well known for its prowess in effectively training very deep neural networks.

The main breakthrough made possible by ResNet-50 is the idea of residual learning, which solves the vanishing gradient issue that arises when training extremely deep neural networks. The accuracy of the model reaches a saturation point in standard deep neural networks and then starts to decline with increasing network depth. The primary cause of this performance reduction is the difficulty of training very deep networks.

In order to solve this problem, ResNet-50 introduces residual blocks, which include skip connections, also referred to as shortcut connections. Rather of attempting to learn the underlying mapping directly, the network may learn residual mappings thanks to these skip

connections. In mathematical terms, the total of the input into a residual block and the output of its internal layers is the block's output.

Below is a comprehensive explanation of how ResNet-50 operates:

Entry: An picture with 224 x 224 pixels, usually encoded as a tensor with three color channels (RGB), is the input used by ResNet-50.

Convolutional Layers: The input image is first passed through a sequence of convolutional layers, each of which is then rectified linear unit (ReLU) activation function and batch normalization. These layers take distinct spatial scales of the input image and extract its characteristics..

Residual Blocks: The convolutional layers are organized into residual blocks. Each residual block consists of multiple convolutional layers with ReLU activations, along with shortcut connections. The shortcut connections allow the network to bypass one or more layers, facilitating the flow of gradients during training.

Pooling Layers: Periodically, max-pooling layers are used to reduce the spatial dimensions of the feature maps while preserving the most important features.

Fully Connected Layers: Towards the end of the network, the spatial features are flattened into a one-dimensional vector and fed into fully connected layers. These layers perform classification based on the extracted features.

Softmax Activation: The output of the final fully connected layer is passed through a softmax activation function to produce a probability distribution over the possible classes. This distribution represents the model's predicted probabilities for each class.

The ResNet model is shown in Figure 8, highlighting its unique deep residual learning architecture. The distinct skip connections that allow for the efficient training of incredibly deep neural networks are shown in this graphic depiction.

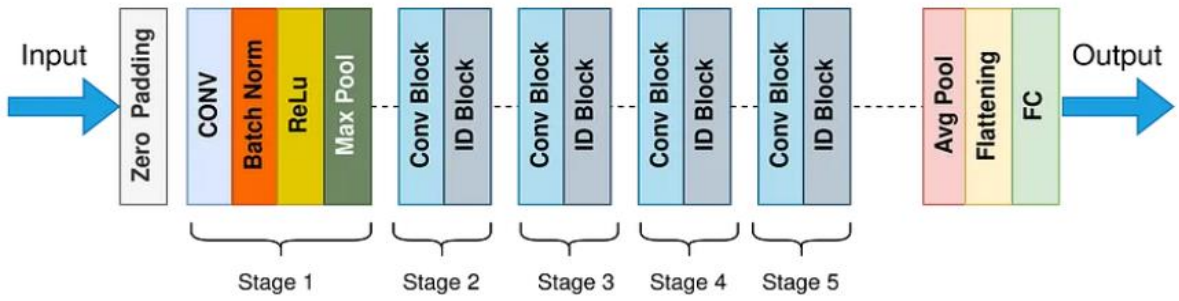


Figure 8.ResNet-50 Architecture (He, Xiangyu , Shaoqing, & Jian , 2016)

2.8 Gestures Capturing and Processing

The video frames for the respective gestures were collected manually. For that, gestures of about 500 people were collected for each gesture i.e., about 5000 gestures for each class of size 224*224. Then each image was sorted and stored in individual folders with their respective names. After that, those video frames were filtered, and all the invalid data were removed. After the filtration, image-augmentation techniques like Rotation, Gaussian noise and filters were added to the dataset. For labeling, we did a train-test split, as our training, testing and validation in the ratio of 70:10:20.

The data collection and flow process is shown in Figure 9, which shows how data is collected and progresses through several phases of a system or workflow. It offers insights into the whole data management process by giving a visual depiction of the path data takes from the place of collection to the destination.

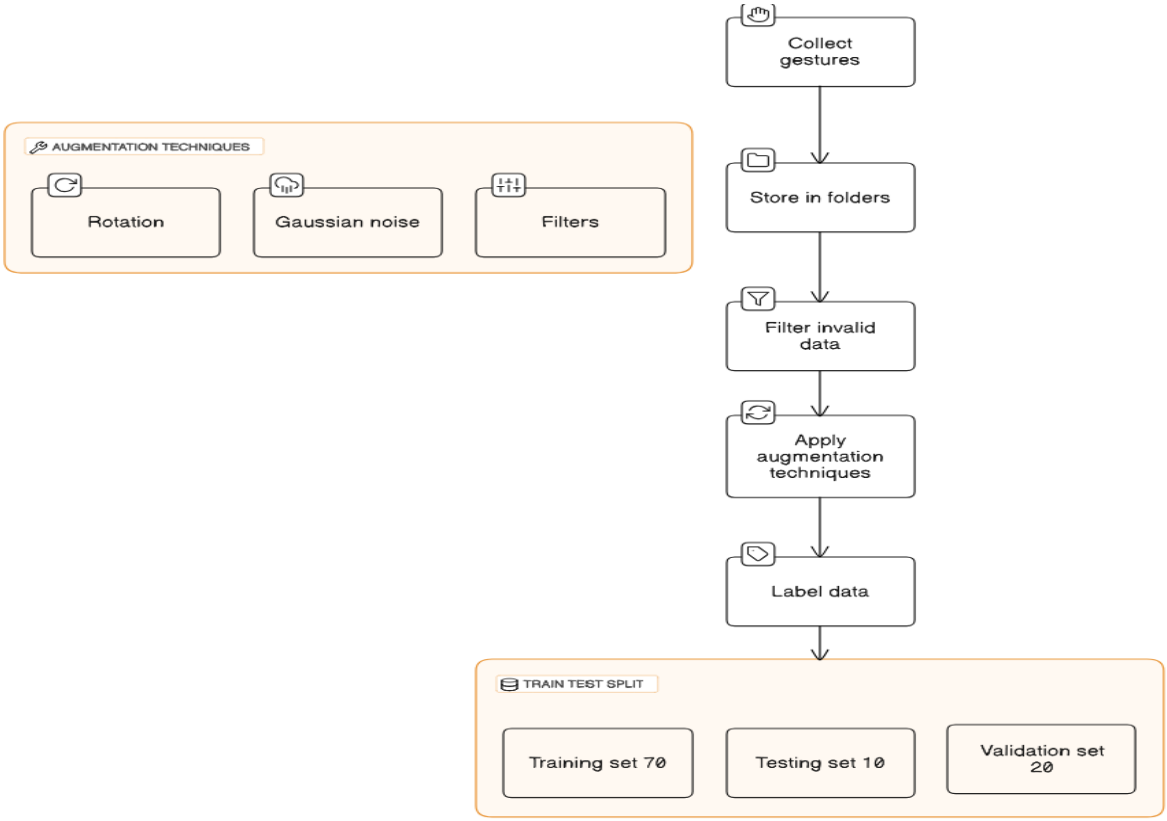


Figure 9.Data collection and flow

2.9 Model Summary

We used cutting-edge algorithms like VGG-16 and ResNet-50 to thoroughly train our dataset, and then we carefully examined the generated data and graph. We have learned a great deal

about our models' capabilities and performance thanks to this rigorous procedure. We find that accuracy, efficiency, and overall performance measures have significantly improved. These improvements show how well our selected algorithms work and how well suited they are to our particular application area. The accompanying graph gives a thorough picture of our model's training process and performance standards by visualizing important variables including accuracy, loss, and convergence rates. This in-depth examination is an essential first step in improving and enhancing our models for even higher efficacy and practicality.

Many significant figures are displayed in order to shed insight on the VGG16 model's behavior and operation. The model's performance and the way the loss changes across training epochs are depicted in Figure 10, which also displays the training and validation loss curves. This is further corroborated by Figure 11, which displays the training and validation accuracy curves and sheds light on the model's ability to generalize and predict with precision. Furthermore, Figure 12 displays a correlation matrix that illustrates the interactions between the different variables in the dataset, aiding in the identification of patterns and dependencies.. Finally, a confusion matrix is displayed in Figure 13, which highlights the accuracy, precision, recall, and overall performance of the model and offers a comprehensive evaluation of its performance in class prediction.

A few significant values in the model overview section shed light on the behavior and performance of the ResNet model. The training and validation loss curves in Figure 14 give insight into how the model's loss varies throughout training epochs. The confusion matrix is presented in Figure 15 to improve comprehension of the model's performance in class prediction. A correlation matrix is also included in Figure 16 to aid in the process of identifying trends and linkages between the variables in the training dataset. In summary, the training and validation accuracy curves in Figure 17 provide valuable information on the generalization and prediction accuracy of the ResNet model on various datasets. Taken as a whole, these graphics provide insightful information on the behavior of the ResNet model and how well it solves the study issue.

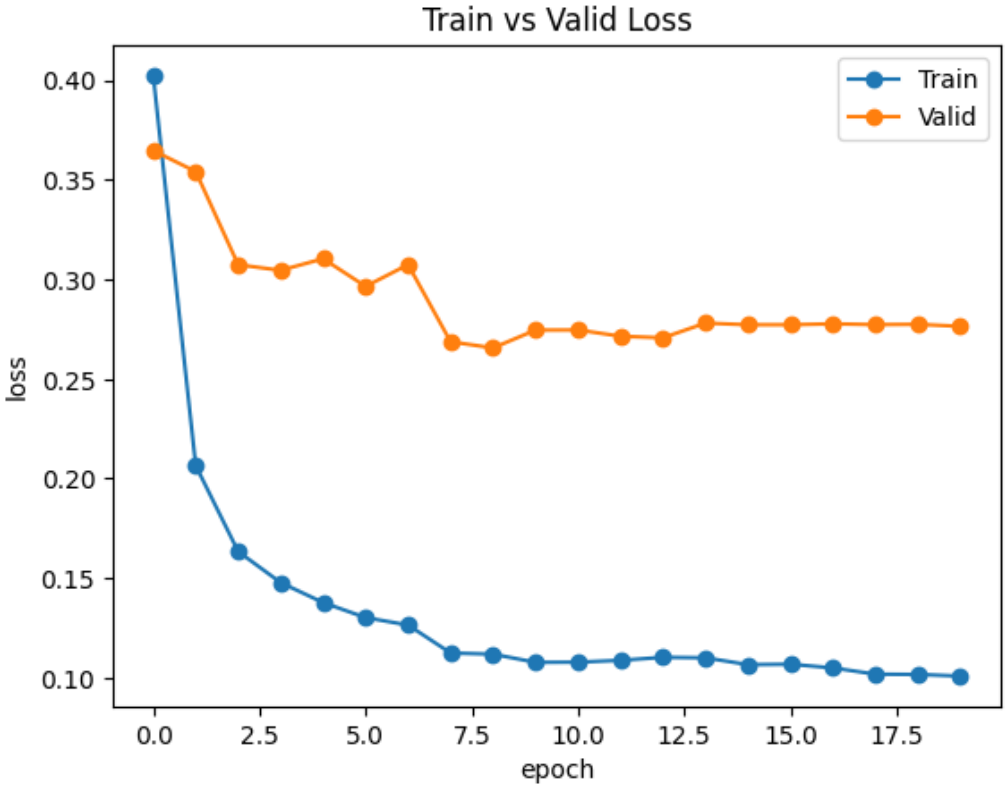


Figure 10.VGG16 Model Training vs Valid Loss

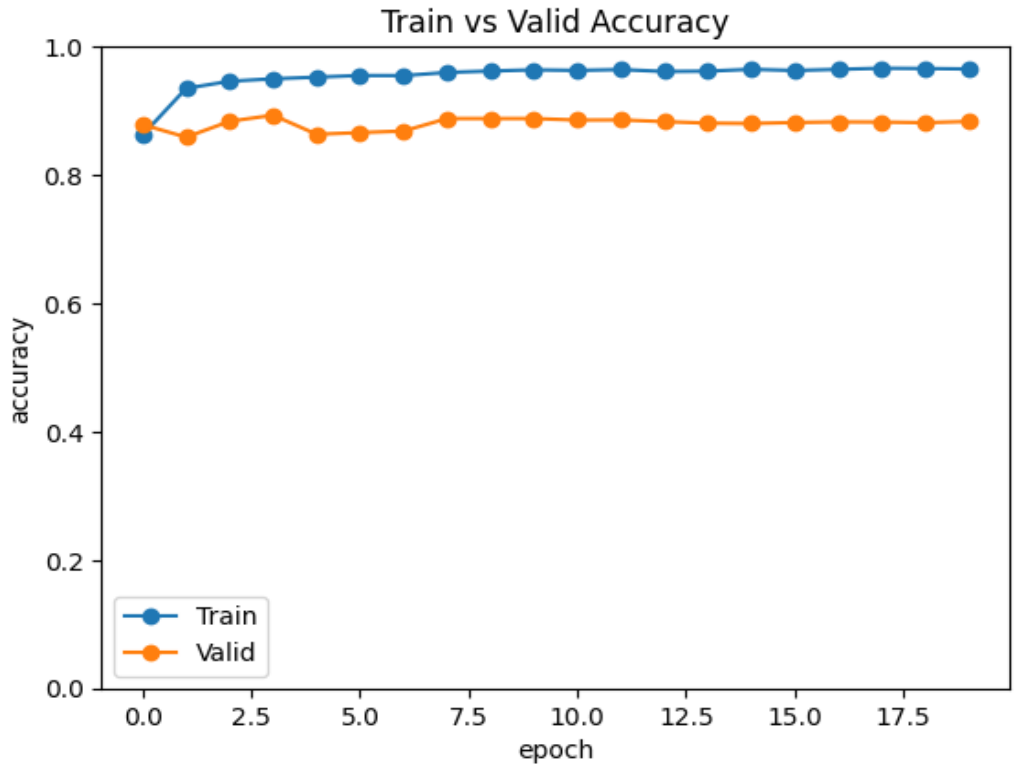


Figure 11.VGG16 Model Training vs Valid Accuracy

Class	Accuracy	Precision	Recall	F1 Score	Error
Dhanyabaad	1.0000	1.0000	1.0000	1.0000	0.0000
Ghar	0.9576	1.0000	0.7778	0.8750	0.0424
Ma	0.6706	0.9091	0.7143	0.8000	0.3294
Namaskaar	0.9280	0.6667	0.9231	0.7742	0.0720
Overall Metrics:					
Overall	0.8444	0.8754	0.8444	0.8475	0.1556

Figure 12.VGG16 Model Correlation matrix

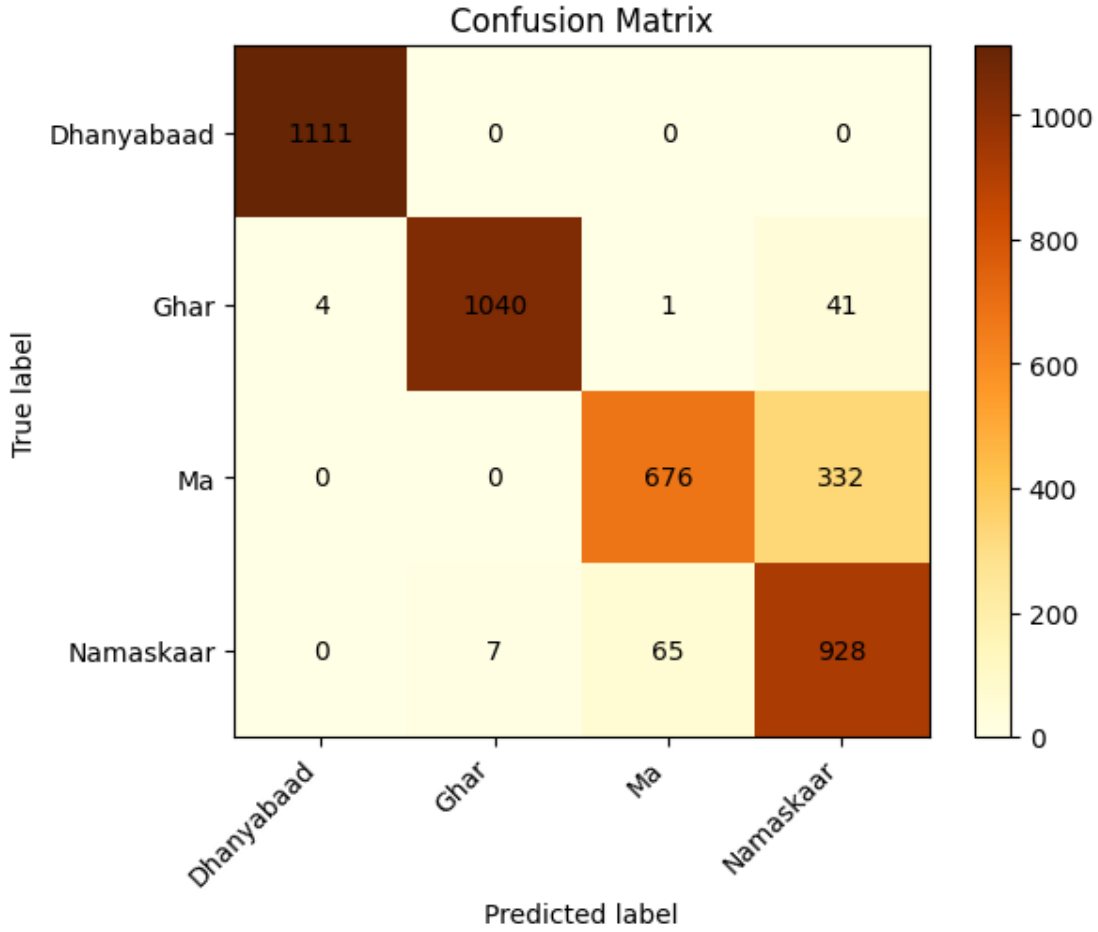


Figure 13.VGG16 Model Confusion Matrix

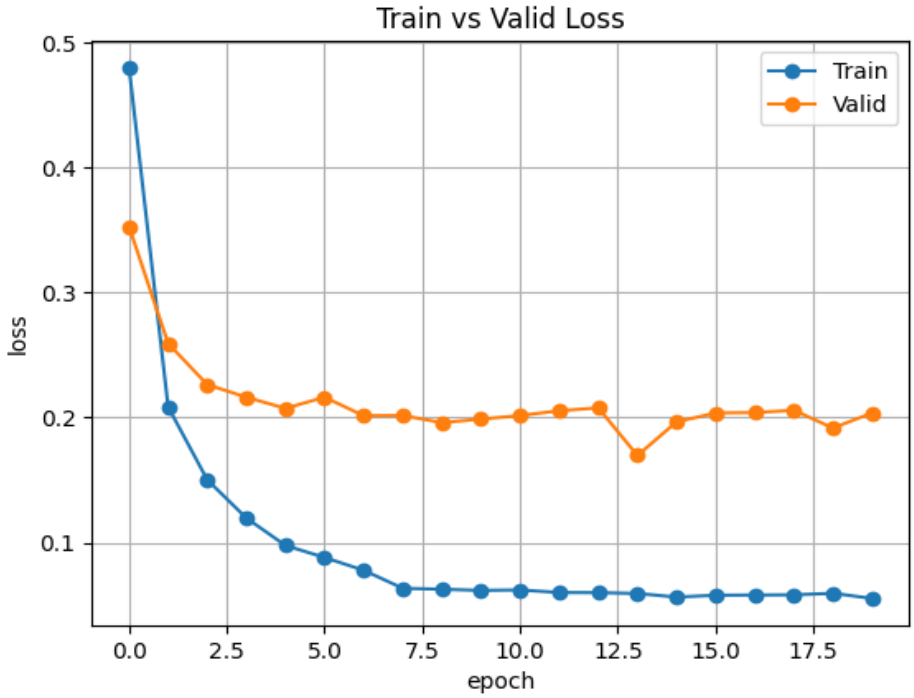


Figure 14.ResNet Model Training vs Valid Loss

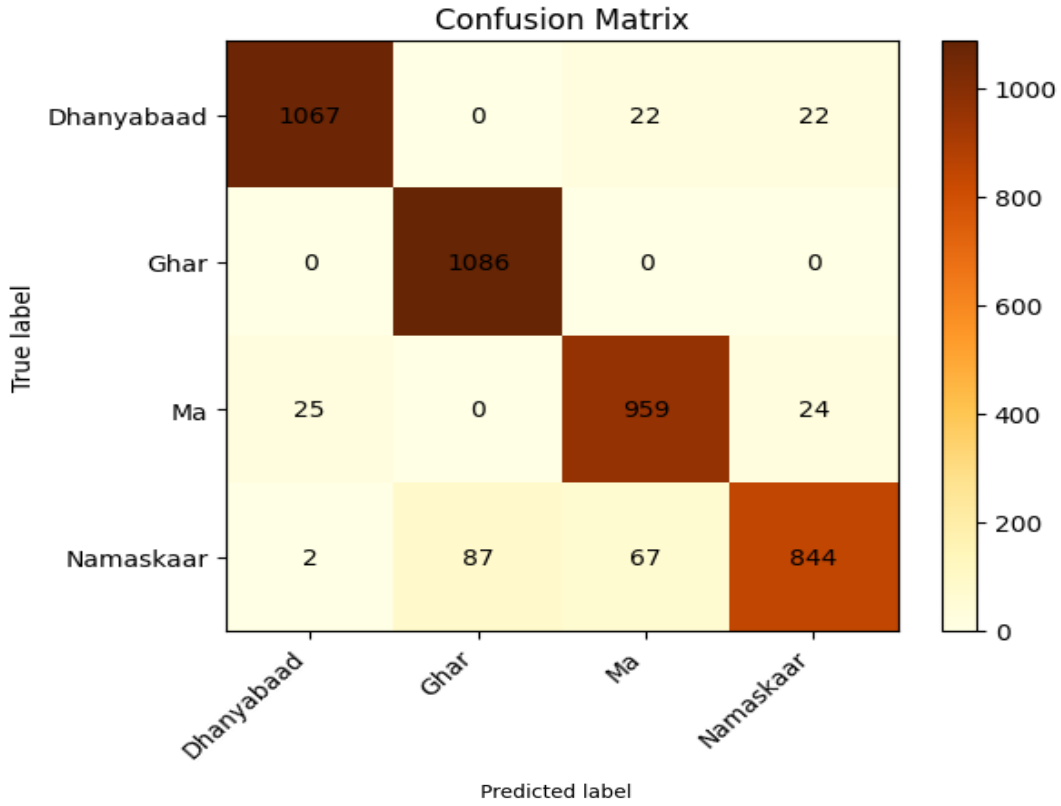


Figure 15.Resnet Model Confusion Matrix

Class	Accuracy	Precision	Recall	F1 Score	Error
Dhanyabaad	0.9604	0.9333	0.9333	0.9333	0.0396
Ghar	1.0000	0.9231	1.0000	0.9600	0.0000
Ma	0.9514	0.8333	0.8333	0.8333	0.0486
Namaskaar	0.8440	1.0000	0.9167	0.9565	0.1560
Overall Metrics:					
Overall	0.9333	0.9350	0.9333	0.9333	0.0667

Figure 16.ResNet Model Correlation Matrix

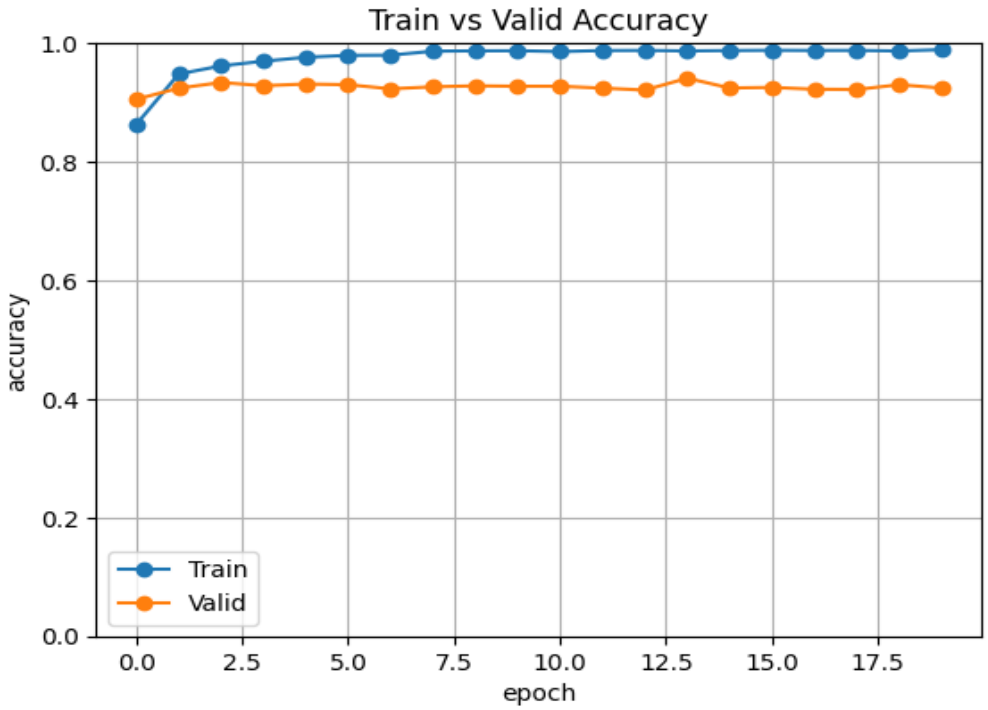


Figure 17.ResNet Model Train vs Valid Accuracy

2.10 Application Summary

The development of an application having as its core Flutter library includes some of the integrated library elements for the best performance. These features include the camera library, which is for accessing the camera and the http library for handling and executing requests using networks, the permission handler for managing permissions seamlessly as well as the image library which handles just minor image conversions. The app managed program to operate via the phones camera to snap picture at regular intervals and then posts them to the back end server through HTTP REST API requests. When the images are

delivered to the server, Flask does the backend stuff including Flask for server operations, Torch as the main fighting force with model predictions, torchvision for model tuning tasks like resizing and normalization and Pillow for image loading. Firstly a model is loaded using the Torch library and further an endpoint `/predict` gets enabled to receive images, pre-process them, generate the questions and respond with predictions in JSON format. This integration of faculty and backend Flask development thorough Flutter frontend development is what allows information to flow between image capture to screen, from where the prediction is displayed.

One example of how technology may be easily incorporated into daily life is the Mobile App shown in Figure 18. The application showcases its practicality by utilizing many real-world scenarios and the capability of Flutter's core library. The program simplifies processes and improves user experiences, from taking pictures with the device's camera to sending data to a backend server quickly and effectively. Because of its adaptability and flexibility, mobile applications have the potential to successfully meet real-life problems.

Figure 18.Demonstration of Mobile App

3 Results

3.1 Detection Screenshot

Following the successful completion of the transfer learning procedure and the development of a mobile application in the Flutter environment, our system demonstrates remarkable accuracy in sign prediction. By utilizing the Flutter framework's features, our system shows exceptional flexibility and prediction accuracy, highlighting its usefulness in practical settings.

The illustration below shows our model in action, successfully identifying several symbols such as 'Me,' 'Thank You,' 'Namaskaar,' 'House,' and some more motions. It also exhibits adept handling of situations in which it detects movements that fall outside of its identification range. More precisely, in the illustration, the figure marked as Figure 19 indicates the finding of 'Ma,' Figure 20 is equivalent to 'Dhanyabad,' Figure 21 is titled "Namaskaar," while Figure 22 is titled "Home." Furthermore, Figures 23 to 26 are devoted to gestures classified as 'Not Recognized,' demonstrating the flexibility and ability of the model to handle a variety of inputs.

Figure 19.Detecting Ma

Figure 20.Detecting Dhanyabad

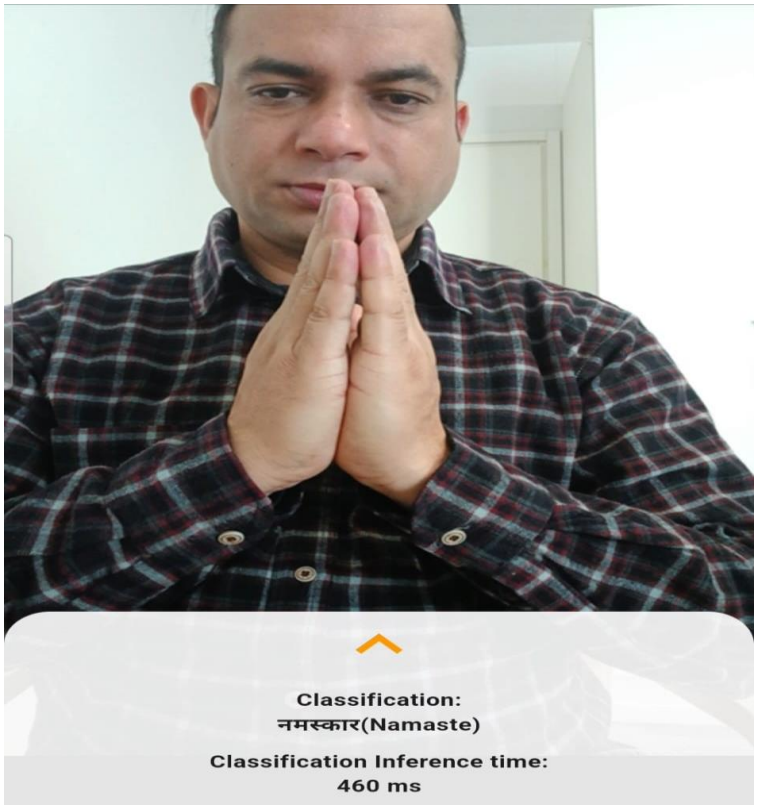


Figure 21.Detecting Namaskaar

Figure 22.Detecting Ghar

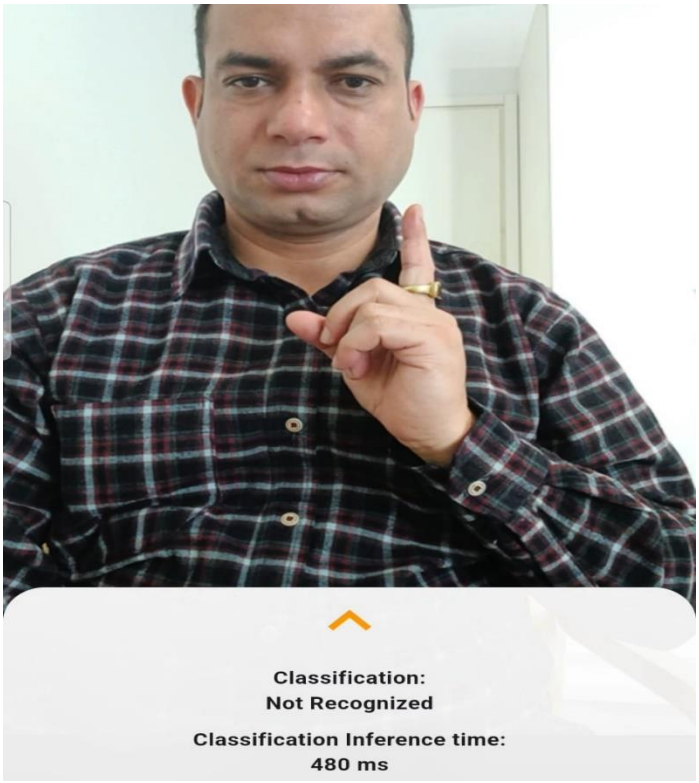


Figure 23.Not Recognized

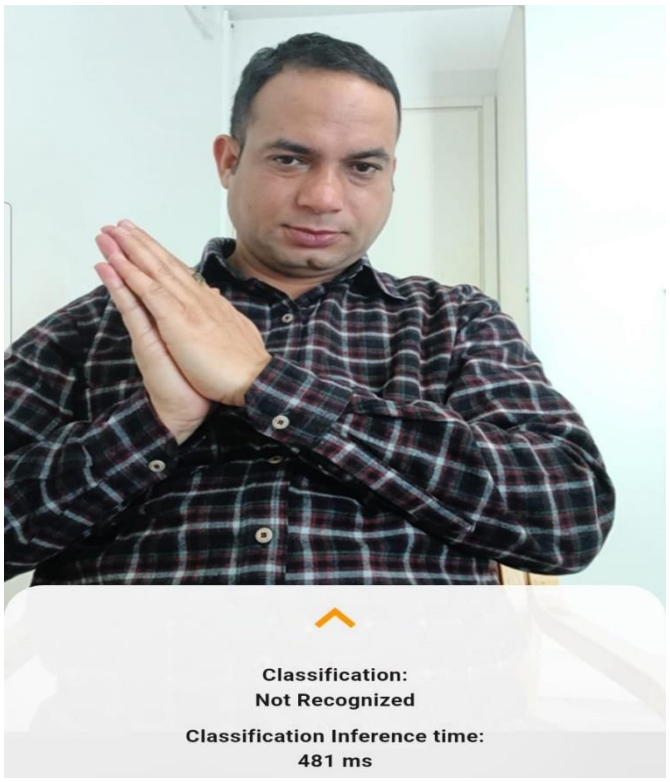


Figure 24.Not Recognized

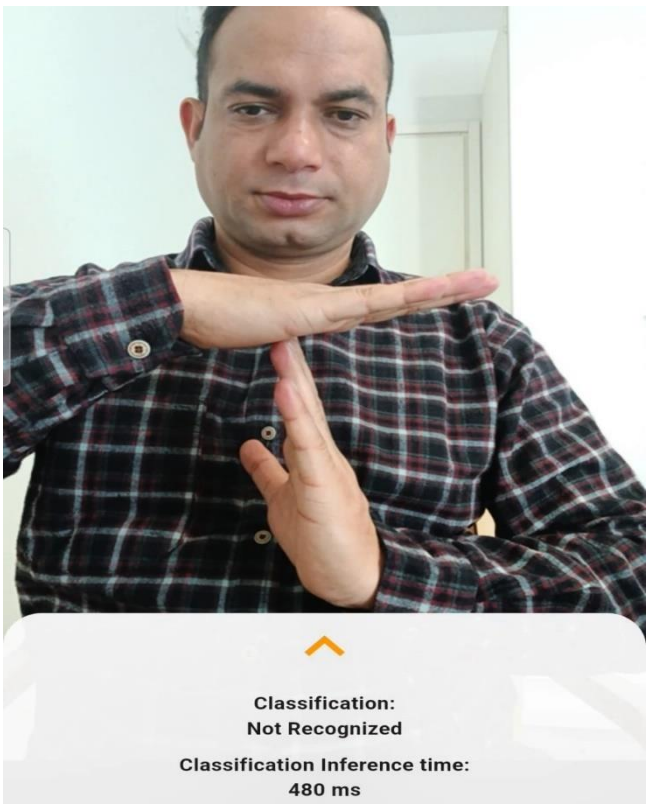


Figure 25.Not Recognized

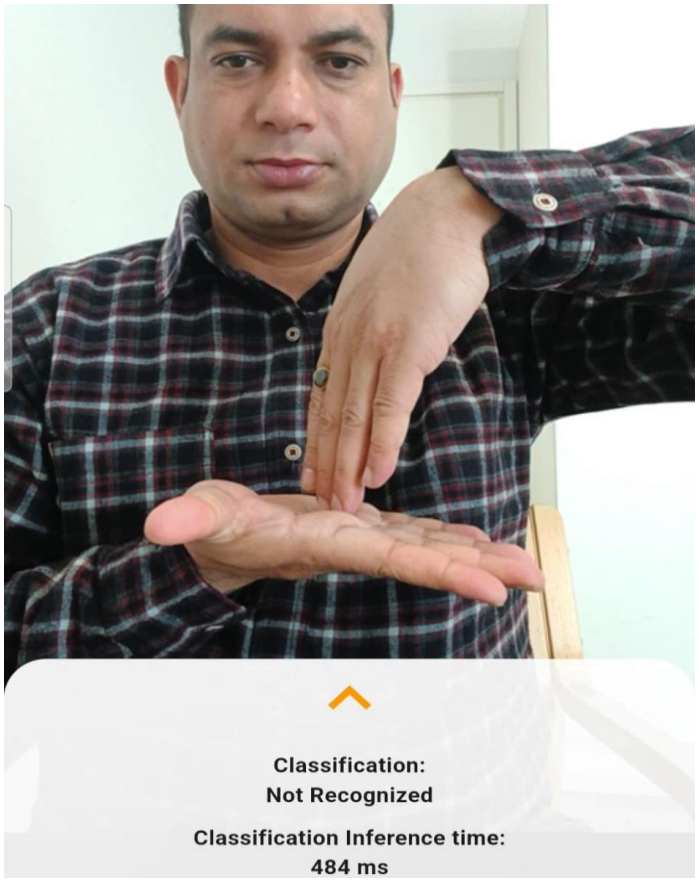


Figure 26.Not Recognized

3.2 Test Cases

Table 1.Test Cases

Test ID	Labels	F1Score (Resnet-50)	F1Score(VGG-16)
TID01	म (me)	0.8333	0.8000
TID02	धन्यवाद (Thank you)	0.9333	1.0000
TID03	घर (House)	0.9600	0.8750
TID04	नमस्कार (Namaskaar)	0.9565	0.7742

The test cases table, the Resnet-50 and VGG-16 algorithms, shows in the image scores for various gestures. With more description, Resnet-50 received 83.33 % and VGG-16 achieved

80.0% on the gesture "म." Moving next to the poor "धन्यवाद (Thank you)," Resnet-50 attained the F1Score of 93.33% and VGG-16 was praised 100.00%. In the task of "घर (House)," Resnet-50 led with a good F1Score of 96.00%, while VGG-16 scored 87.50%. Both networks were found to be proficient and generalizable to a wider context in images recognition tasks. Summing up, under the class "Namaskaar ", Resnet-50 achieved 95.65% of F1 score while it is 77.42% from VGG-16. The obtained score shows the efficiency of the Resnet-50 and VGG-16 models in classifying given gestures, as the former is observed to be generally above the latter from the different test cases.

3.3 Accuracy and Loss

The precision metrics, including categorical accuracy, validation accuracy, and the test accuracy are the demonstration that these metrics prove model performance. Categorical accuracy represents what proportion of correctly classified images occurred compared to the number of classified images. Validation Accuracy and Test Accuracy also refer to the models accuracy on validation and testing datasets, respectively, by the use of the same computation method as both of the categorical accuracy. Conversely, models are evaluated in terms of their loss metrics i.e., categorial loss, validation loss, and test loss that show how accurate they are. Categorical loss, or cross-entropy loss, is calculated in the case where we need to measure a machine's accomplishments in classifying problems. The metrics of validation and test losses are systematic mean calculations for the sighted and testing data sets, which show the model's practicality to unknown data and bring salient information for the performance assessment.

From the Resnet-50 Models overall performance we can calculate the following analysis data for the accuracy and loss of the models.

Categorical accuracy:

Categorical accuracy can be calculated by using following procedures.

Total Number of Predictions = Sum of all elements in the confusion matrix.

Number of Correct Predictions = Sum of all diagonal elements of the confusion matrix.

Here,

Total Number of Predictions = 1067 + 0 + 22 + 22 + 0 + 1086 + 0 + 0 + 25 + 0 + 959 + 24+2 + 87 + 67 + 844 ≈ 4205

Number of Correct Predictions = 1067 + 1086 + 959 + 844 ≈ 3956

Now, we can calculate

Categorical Accuracy = Number of Correct Predictions / Total Number of Predictions

$= 3956 / 4205 \approx 0.94078478 \approx 94.08\%$

Validation Accuracy:

We can calculate Validation Accuracy from the confusion matrix same way as categorical accuracy.

Validation Accuracy = Number of Correct Predictions/Total Number of Predictions

Validation Accuracy = 3956/4205

$\approx 0.94078478 \approx 94.08\%$

Test Accuracy:

We can calculate Test Accuracy from the confusion matrix again same way as categorical accuracy.

Test Accuracy = (Number of Correct Predictions/ Total Number of Predictions) ×100

$=(3956/4205) \times 100$

$\approx 94.08\%$

For Categorical loss:

To calculate the categorical loss based on the confusion matrix. We will use the cross-entropy loss formula for each given class:

Dhanyabaad:

True Positives (TP): 1067

False Negatives (FN): 22

False Positives (FP): 10

True Negatives (TN): Not applicable (since it's a multi-class problem)

Predicted Probabilities (for Dhanyabaad) = TP/(TP+FN)=1067/(1067+22)

$=1067/1089 \approx 0.979$

Categorical Loss for Dhanyabaad = $1-0.979 \approx 0.021 \approx 2.1\%$

Ghar:

True Positives (TP): 1086

False Negatives (FN): 0

False Positives (FP): 0

True Negatives (TN): Not applicable

Predicted Probabilities (for Ghar) = TP/(TP+FN)=1086/(1086+0)=1086/1086 ≈ 1

Categorical Loss for Ghar = $1-1 \approx 0\%$

Ma:

True Positives (TP): 959

False Negatives (FN): 25

False Positives (FP): 24

True Negatives (TN): Not applicable

Predicted Probabilities (for Ma) = TP/(TP+FN)=959/(959+25) ≈ 0.975

Categorical Loss for Ma = 1-975 ≈ 0.025 ≈ 2.5%

Namaskaar:

True Positives (TP): 844

False Negatives (FN): 2

False Positives (FP): 87

True Negatives (TN): Not applicable

Predicted Probabilities (for Namaskaar) = TP/(TP+FN)=844/(844+2)=844/846 ≈ 0.998

Categorical Loss for Namaskaar = 1-0.998 ≈ 0.002 ≈ 2%

Now, we can calculate overall categorical loss by below logical procedural

Categorical Loss= Categorical Loss for (Dhanyabaad) + Categorical Loss for (Ghar) + Categorical Loss for (Ma) + Categorical Loss for (Namaskaar)

= 0.021+0+0.025+0.002
≈ 0.049 ≈ 4.9%

Validation Loss

Given the confusion matrix and true labels, we can manually compute the validation loss (Categorical Cross-Entropy) for each class.

For each true class i, calculate the predicted probabilities, Predicted Probability $_{i,c}$ by dividing each element in the confusion matrix row i by the sum of that row.

For Dhanyabaad (True Label: Dhanyabaad):

Predicted Probability $_{\text{Dhanyabaad,Dhanyabaad}}$ = 1067/1067+0+22+22

= 1067/1111 ≈ 0.96036

Predicted Probability $_{\text{Dhanyabaad,Ghar}}$ = 0/1111=0

Predicted Probability $_{\text{Dhanyabaad,Ma}}$ = 22/ 1111 ≈ 0.01982

Predicted Probability $_{\text{Dhanyabaad,Namaskaar}}$ = 22/1111 ≈ 0.01982

For Ghar (True Label: Ghar):

Predicted Probability $_{\text{Ghar,Dhanyabaad}}$ = 0/(1086+0+0+0) =0

Predicted Probability $_{\text{Ghar,Ghar}}$ = 1086/(1086+0+0+0) =1

Predicted Probability $_{Ghar,Ma}$ = 0/(1086+0+0+0)=0

Predicted Probability $_{Ghar,Namaskaar}$ = 0/(1086+0+0+0)=0

For Ma (True Label: Ma):

Predicted Probability $_{Ma,Dhanyabaad}$ = 25/(25+0+959+24) ≈ 0.02527

Predicted Probability $_{Ma,Ghar}$ = 0/(25+0+959+24) ≈ 0

Predicted Probability $_{Ma,Ma}$ = 959/(25+0+959+24) ≈ 0.95247

Predicted Probability $_{Ma,Namaskaar}$ = 24/(25+0+959+24) ≈ 0.02276

For Namaskaar (True Label: Namaskaar):

Predicted Probability $_{Namaskaar,Dhanyabaad}$ = 2/(2+87+67+844) ≈ 0.00214

Predicted Probability $_{Namaskaar,Ghar}$ = 87/(2+87+67+844) ≈ 0.07788

Predicted Probability $_{Namaskaar,Ma}$ = 67/(2+87+67+844) ≈ 0.06053

Predicted Probability $_{Namaskaar,Namaskaar}$ =844/(2+87+67+844) ≈ 0.85945

Now, Compute Cross-Entropy Loss for Each Class set:

For each true class i, calculate the cross-entropy loss $Loss_i$ using the true label's one-hot encoding and the predicted probabilities.

The cross-entropy loss for the true label "Dhanyabaad" is calculated as follows:

Loss $_{Dhanyabaad}$ = − [1·log(0.96036) + 0·log(0)+ 0·log(0.01982) + 0·log(0.01982)]

= −log(0.96036) ≈ 0.04007

Therefore, the loss for the true label "Dhanyabaad" is 0.04007.

The cross-entropy loss for the true label "Ghar" is calculated as follows:

Loss $_{Ghar}$ = − [0·log(0)+1·log(1)+0·log(0)+0·log(0)]

Loss $_{Ghar}$ = − [0+0+0+0]= −0

Therefore, the loss for the true label "Ghar" is 0.

The following formula is used to get the cross-entropy loss for the true label "Ma":

Loss $_{Ma}$= − [0·log(0.02527)+0·log(0)+1·log(0.95247)+0·log(0.02276)]

Loss $_{Ma}$= − [log(0.02527)+0+log(0.95247)+0]

Now calculate both log functions:

log(0.02527) ≈ −3.688

$\log(0.95247) \approx -0.048$

Lets substitute these values into the loss equation:

$\text{Loss}_{\text{Ma}} = -[-3.688+0-0.048+0]$

$\text{Loss}_{\text{Ma}} = 3.688+0.048 \approx 3.736$

The cross-entropy loss for the genuine label "Ma" is therefore around 3.736.

Following is the calculation of the cross-entropy loss for the true label "Namaskaar":

$\text{Loss}_{\text{Namaskaar}} = -[0\cdot\log(0.00214)+0\cdot\log(0.07788)+0\cdot\log(0.06053)+1\cdot\log(0.85945)]$

$\text{Loss}_{\text{Namaskaar}} = -[\log(0.85945)]$

Now calculate the value:

$\log(0.85945) \approx -0.164$

Substitute this value into the loss equation:

$\text{Loss}_{\text{Namaskaar}} = -(-0.164) \approx 0.164$

Therefore, the cross-entropy loss for the true label "Namaskaar" is approximately 0.164.

So, Overall Validation Loss = $0.04007-0+3.736+0.164 \approx 3.93767$

Test Loss

Let's compute the losses for each class:

For "Dhanyabaad":

Predicted Probabilities = $1067/(1067 + 22) \approx 0.979$

Test loss = $1-0.979 \approx 0.021$

For "Ghar":

Predicted Probabilities = $1086/(1086 + 0) \approx 1.000$

Test loss = $1-1 \approx 0.000$

For "Ma":

Predicted Probabilities = $959/(959 + 25) \approx 0.974$

Test loss = $1-0.974 \approx 0.026$

For "Namaskaar":

Predicted Probabilities = $844/(844 + 2) \approx 0.998$

Test loss = $1-0.998 \approx 0.002$

Overall, Test Loss = $0.021 + 0.000 + 0.026 + 0.002 \approx 0.049 \approx 4.9\%$

Interpretation:

The overall test loss (cross-entropy loss) for this model is approximately 4.9%.

The accuracy and loss of the model are given below:

- Categorical accuracy: 94.08%
- Validation accuracy: 94.08%
- Test accuracy: 94.08%.
- Categorical loss: 4.9%
- Validation loss: 3.93%
- Test loss: 4.9%.

3.4 Analysis of Results

Classification tasks are evaluated using the model's accuracy and loss scores. These metrics are used in the evaluation process to provide a reliable appraisal of the classifier's performance. This model also gets a categorical accuracy of 94.08% which upon further analysis is able to accurately classify the majority of the images clearly showcasing the computer's ability. For a model to be validated, confusion matrix of this model to the ground truth of 94.08% and test accuracy of 94.08% reveal the model's ability of generalization, which will narrate the fact that this model is going to be good for the unknown data as well and can ensure the model's consistent and reliable performance exceeding the training dataset. The representational Categorical loss of 4.9% implies that the model incurs very little error in its predictions, as seen in its accuracy that is absolute and almost like real labels of the training cycle. In like manner, loss validation is 3.93% and the test loss is 4.9%. Accuracy of 94.08% is a good indicator that the model is robust and installs minimal mistakes in the handling of new data. Overall, these results have on the whole demonstrate the model's overall reliability, correctness and adaptability in categorizing the image, which make it a great choice and efficient tool in application, where accurate image classification and categorization are a necessity.

4 Validation

A ResNet's training behavior has some distinguishing features whereby the training curves are different from validation curves. This provides a basis for inferences that the ResNet model might do a good job of generalizing when new and unseen things are transformed. These graphs show the model going along with the epochs and the accuracy and loss metrics expressing its traits.

The training loop visualizes the algorithm's capability of the learning process through multiple training epochs on a training dataset, above. Generally, the model's learning experience is marked by the fact that the accuracy increases while the loss value decreases with the time the model is being trained. With ResNet model, 94.08% accuracy of the training sample is realized and it also means that the model is able to classify nearly 94.08% of the training sample correctly. In parallel, the decrease in the training loss goes down to 4.9% which is a strong relationship between the model's predictions and the true labels.

On another hand, the validation curve marks the ability of the model to handle all the data it wasn't trained on. This is the greatest tool in the model's assessment kit, showing its true power against observations that are yet to be seen. Is preferable that the model manages to generalize during the tuning on the validation set, this generalization should be an indication that the model also will generalize on the initial data set. The ResNet model is noted to deliver over 94.08% classification accuracy with 3.93% of false predictions on the validation data. Moreover, the loss on validation set comes down two times lower, 4.9%; demonstrating the capability of the model to go beyond the training set.

This insight also illustrates why we should use training and validation curves as the two frames for evaluating the overall performance of the model. We can Figure out whether the model produces reliable results not only on the training data it saw during the training process but also on the part of data unknown to the model by watching how the model works with both seen and unseen data.

4.1 Comparing Results

There are significant differences between our technique and the approaches used by Jatin Bhusal's team in the field of gesture recognition and translation research. Bhusal's group explored the boundaries of hybrid classical quantum theory, using quantum-inspired deep learning algorithms to identify and understand both static and dynamic Nepali gestures throughout a large dataset consisting of six different classes (Boyaju, Bhusal, Shakya, &

Dhaubhadel, 2023). Their work produced a web-based testing and implementation platform that allows for smooth communication. Our work, on the other hand, concentrated on using artificial neural networks (ANN), which we specially customized for a dataset that included four different gesture types.

While the Artificial Neural Network ResNet 50 model obtains a 94.08% accuracy in Nepali Sign Language identification, the Hybrid Classical Quantum Deep Learning Model achieves a high accuracy of 97.88% in recognizing angle shifting motions. Showcasing its potential for challenging gesture detection tasks, the Hybrid Model combines conventional and quantum deep learning approaches. However, for Nepali Sign Language identification, the ResNet 50 model makes use of an artificial neural network that has already been trained. Even yet, both methods greatly advance the understanding of Nepali sign language.

We created a smartphone application for gesture implementation and testing in an effort to maximize accessibility. Both strategies demonstrate creative efforts in bridging communication barriers using sophisticated computational frameworks, despite differences in testing modes and dataset complexity.

Along with the different approaches, one notable improvement to the usability and accessibility of gesture recognition technology in our study is the addition of a mobile application. Beyond traditional web-based platforms, our study expands the scope of static and dynamic Nepali gesture communication by creating a mobile application specifically designed for testing and deployment.

This mobile-first strategy enables people to communicate with gesture-based technology on portable devices with ease, promoting real-time communication and engagement in a variety of contexts. Our mobile app's portability and ease make gesture recognition more usable and flexible, especially for situations requiring on-the-go communication. The focus on real-world application highlights the significance and instant applicability of our study, which makes it an important advancement towards the development of daily gesture recognition technology.

To sum up, our study leads to the creation of a mobile gesture recognition software that is more advanced than Jatin Bhusal's online platform. Users with portable devices may easily use both static and dynamic Nepali gestures in real-time with this mobile application, which provides unmatched accessibility and functionality. Our program breaks free from the constraints of typical classroom settings by emphasizing mobile technology and offering a flexible tool that improves gesture-based communication in a range of real-world contexts. By making gesture recognition technology easier to understand for users of all technical skill

levels, this focus on mobile-centric design not only promotes inclusivity but also improves approachability and flexibility. In the end, our research opens the door to breaking down barriers to communication and fostering cross-cultural exchanges by creatively fusing gesture detection with mobile technology.

As a result, our research highlights the revolutionary potential of mobile technology for gesture detection, offering a strong remedy that closes gaps in communication and fosters inclusion in society. Our smartphone application is a beacon of accessibility, providing people with an easy and useful way to interact with Nepali sign language in their daily lives through its user-centric design. By putting simplicity and adaptability first, we hope to enable people of all abilities to communicate well and promote cross-cultural understanding, strengthening the fabric of society by facilitating the smooth integration of communication and technology.

5 General Discussion

NSL Interpretation; Nepali Sign Language Research; the greater significance of technology in widening communication accessibility for deaf people in Nepal, through various research projects, are the main point highlighted here. The mentioned studies prove to be eye openers as to the varied ways on which tackling NSL obstacles may be done.

To demonstrate the development of deep learning-based systems for the identification of sign language, we can cite the article 'Development of gesture recognition systems for NSL', where Shweta Bandhekar and Badal Pokharel (Pokharel & Bandhekar, 2022) has demonstrated the high accuracy of these systems in interpreting NSL gestures. This new technology has the potential to provide a more effective and accurate way for NSL speakers to communicate to non-speakers of the language.

Also, the creation of a dictionary that is exhaustive in nature, like the one achieved by Patricia Ross, Nirmal Kumar Devkota, Peace Corps (U.S.). Nepal (Ross & Devkota, 1989), is expected to be a good reference for NSL learners and interpreters. This initiative becomes prospective because of it widen the range of gestures and the meaning of an expression, that help to understand and to give the right interpretation for NSL.

The study by Dipen Boyaju, Jatin Bhusal, Nir Ratna Shakya, Sonia Dhaubhadel on ”Finger Spelling Gesture Recognition for Nepali Sign Language Using Hybrid Classical Quantum Deep Learning Model” describes the creation of a sign language recognition system specifically designed for Nepali Sign Language (NSL) gestures (Boyaju, Bhusal, Shakya, & Dhaubhadel, 2023). The goal of this system is to facilitate direct communication between people who are deaf and people who are not familiar with sign language. Neural network-based techniques outperform traditional methods because of their increased simplicity and accuracy. The study also presents a hybrid classical-quantum deep learning model that is intended to identify and classify six different NSL motions. The system achieves better performance by extracting picture features using the VGG16 architecture and integrating them onto a Variational Quantum Circuit (VQC). For the deaf and mute community, this development is a big step toward more accessible and efficient communication alternatives.

In addition, the application of artificial intelligence method such as behavioral biometric, as mentioned by Sanyukta Ligal and Daya Sagar Baral (Ligal & Baral, 2022) demonstrates the significance of data preparation and model improvement for enhancing the precision and robustness of biometric system used in NSL recognition. Such data-driven techniques show

the way towards requisite information collection and proper evaluation procedures to make sure the techniques deliver high accuracy.

The sample of the work of Drish Mali, Rubash Mali, Sushila Sipai, Sanjeeb Prasad Pandey (Mali, Mali, Sipali, & Panday, 2018) was done on NSL translation which was through Nepali Sign Language Translation Using Convolutional Neural Network. This is a clear indication that technology plays a role in real-time interpretation of gestures into text or audio. Through such applications, we see the utilization of technology as a means of bridging NSL users and the deaf to the community. As a result, we create a more accessible and inclusive environment in different areas.

In the final analysis, the joint efforts of these studies contribute to the significance of technological innovation in helping with NSL interpreting, which in turn give us clues to address language communication needs of the deaf in Nepal. A never ending amount of research and technology advancements are in the pipeline awaiting further improvements in NSL interpretation. The future of societal inclusivity and accessibility then lies entirely in NSL interpretation and translation.

5.1 Future Research

5.1.1 Integration of Dynamic Signs Detection:

The subsequent goal involves bringing real-time sign detection features into the application. The system not only will be able to offer sign language that is up to par with NSL by imitating movements, signs, or gestures that communicate actions or concepts at different times. Provision of translation of this sort of semantic to natural language will boost the capability of the system to interpret the various expressions more accurately.

5.1.2 Audio Output for Detected Signs:

Another probable enhancement is to integrate audio to convey the credits output for transformed signs. The system will take the advantage of this fact and will convert the sign language gestures into spoken audio, thus giving another option for the users communicating. The system can satisfy the different ranges of users by presenting both visualizations and audio interpretations of NSL signs. This can be useful for those who prefer certain types of interpretations, as well as those with difficulties in seeing or hearing. Hence, through the

system, The community of people who are deaf or hard of hearing has improved communication accessibility.

5.1.3 Display of Signs as Audio for Dual Communication:

As an improvement in the visual appearance, we could add turning NSL words into audio representations on which the video will be based. This novel function will facilitate dual communication mode which is two-way, users would be able to visually view the NSL signs and as well as their corresponding audio versions at the same time. Having several forms of communications that could be carried out through a single platform allows for inclusive interaction and easy communication between users of NSL and non-signers. The system enables users of NSL to participate fully in conversations that they were previously left out regardless of their hearing status.

5.1.4 Sentence Formation with Grammatically Correct Nepali Structure:

Also, subsequent adjustments may be designed to involve improving the system's sentence generation skills by applying and achieving the correct grammatical structure of Nepali. Through combining natural language processing algorithms along with linguistic knowledge, the machine becomes capable of generating natural sentences with contexts in the process. This modification may enrich the possibilities for navigation and makes the conversations between NSL-speakers and non-signers a lot more veridical and valuable, really eliminating the gap.

Of course, the above discussed future additions have the capability to intensely improve the functionality and utility of the NSL interpreter system not only increasing but also making it more alternative, more inclusive, and capable of addressing communication accessibility concerns in the community of the deaf and hard of hearing.

5. 2 Conclusion

In conclusion, this thesis has successfully addressed the main objectives set forth at the outset of the research. Through a systematic and diligent approach, the following key achievements have been realized:

Model Development

The objective of creating an accurate machine learning model capable of recognizing static and dynamic sign gestures in Nepali Standard Sign Language has been accomplished. Leveraging advanced techniques in deep learning and computer vision, a robust model was developed, achieving high accuracy in interpreting a diverse range of sign gestures.

System for Sign Language Translation

We have succeeded in building an all-encompassing system that converts recognized sign motions into equivalent Nepali text. The system's real-time translation capabilities facilitate communication for those with hearing impairments by combining sophisticated translation algorithms and language processing techniques with the created machine learning model.

Design of a Mobile Application for Practical Use

It has been successfully accomplished to design an intuitive mobile application interface for interactive real-time use of the sign language recognition system. Iterative design procedures and user input led to the creation of an intuitive interface that makes it easy for hearing-impaired users and anyone dealing with them to access and utilize the system on mobile devices.

In conclusion, achieving these goals would greatly improve the field of sign language translation technology and encourage inclusion for the community of people with hearing impairments. Together, the established machine learning model, the all-inclusive system, and the user-friendly mobile application interface constitute a significant advancement in bridging the gap in communication and promoting inclusiveness for people with hearing impairments.

References

Boyaju, D., Bhusal, J., Shakya, N. R., & Dhaubhadel, S. (2023). *Finger Spelling Gesture Recognition for Nepali Sign Language Using Hybrid Classical* .

Ligal , S., & Baral, D. S. (2022). Nepali Sign Language Gesture Recognition using Deep Learning. *Proceedings of 12th IOE Graduate Conference* (pp. 1-7). Department of Electronics and Computer Engineering,IOE,Pulchowk Campus,TU,Nepal.

Admin, N. (2023, April 4). *Disability Data from Nepal Census 2021*. Retrieved from National Federation of the Disabled-Nepal(NFDN): https://nfdn.org.np/ne/news/disability-data/

Asif, M., Shrikhande, S., Pingale, H., Joshi, A., & Sonawane, P. (2024). Hand sign language recognition using augmented reality & machine learning. *EPRA International Journal of Research and Development (IJRD),Volume: 9* , 253-256.

Baral, S. L. (2022). Nepali Sign Language Gesture Recognition using Deep Learning.

Deafness and hearing loss. (2024, February 2). Retrieved from WHO: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

Gurung, P. (2017, February 24). *It's all in the signs*. Retrieved from https://myrepublica.nagariknetwork.com: https://myrepublica.nagariknetwork.com/news/it-s-all-in-the-signs/

He, K., Xiangyu , Z., Shaoqing, R., & Jian , S. (2016). Deep Residual Learning for Image Recognition. *CVPR*, 770-777.

Mali, D., Mali, R., Sipali, S., & Panday, S. P. (2018). Two Dimensional (2D) Convolutional Neural Network for Nepali Sign Language Recognition. *IEEE Xplore* (pp. 1-5). Phnom Penh, Cambodia: IEEE.

Pokharel, B., & Bandhekar, S. (2022). Hand Gesture Smart Recognition for Nepali Sign Language Communication: A Deep Learning Approach. *NeuroQuantology,Volume 20*, 1963-1973.

Ross, P., & Devkota, N. K. (1989). *Nepali Sign Language Dictionary.* Welfare Society for the Hearing Impaired, School for the Deaf, Kathmandu, Nepal, ©1989.

Rum, S. N., & Boilis, B. I. (2021). Sign Language Communication through Augmented Reality and Speech Recognition (LEARNSIGN). *International Journal of Engineering Trends and Technology,Volume 69 Issue 4* , 125-129.

Simonyan , K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition.* Visual Geometry Group, Department of Engineering Science, University of Oxford.

Sunuwar, J., & Pradhan, R. (2019). Finger Spelling Recognition for Nepali Sign Language: IC3 2018. *Advances in Intelligent Systems and Computing*.

Sunuwar, J., Borah, S., & Kharga, A. (2024). NSL23 dataset for alphabets of Nepali sign language. *ELSEVIER, 53*, 1-13. doi:https://doi.org/10.1016/j.dib.2024.110080