SIGN LANGUAGE DETECTION USING DEEP LEARNING

BY

Oliur Rahman
ID: 241-25-029

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science and Engineering

Supervised By

Dr. S. M. Aminul Haque
Professor & Associate Head
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised By

Mr. Abdus Sattar
Associate Professor & Director, M.Sc.
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

May 2025

APPROVAL

This Thesis titled **"Sign Language Detection Using Deep Learning"**, submitted by **Oliur Rahman**, ID No: **241-25-029** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of **MSc. in Computer Science and Engineering** and approved as to its style and contents. The presentation has been held on **24-05-2025**.

BOARD OF EXAMINERS

Chairman

Prof. Dr. Sheak Rashed Haider Noori
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Md. Zahid Hasan
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Naznin Sultana
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner

Dr. Md. Zulfiker Mahmud
Professor
Department of Computer Science and Engineering
Jagannath University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. S. M. Aminul Haque, Professor & Associate Head, Department of Computer Science and Engineering, Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Dr. S. M. Aminul Haque
Professor & Associate Head
Department of CSE
Daffodil International University

Co-Supervised by:

Mr. Abdus Sattar
Associate Professor & Director, M.Sc.
Department of CSE
Daffodil International University

Submitted by:

Oliur Rahman
ID: 241-25-029
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to Almighty for His divine blessing making it possible for us to complete the final year project/internship successfully.

We are grateful and wish our profound indebtedness to **Dr. S. M. Aminul Haque Professor & Associate Head**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Machine Learning*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to the **Head of the Department of CSE**, for his kind help in finishing our project and also to other faculty members and the staff of the Department of CSE, Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Sign Language Detection Using with deep Learning is utilized and advance process to detect sign languages by using artificial intelligence (Ai) method like CNN and RNN to recognize hand or arm gesture/posture from image, therefore basically concerned with the design of sign language recognition technology targeted at enhancing communication for the hearing and speech impaired populace. This strongly utilizes deep learning from still pictures, in addition to video sequences of American Sign Language (ASL) gestures. To correctly categorize the gestures, features such as DenseNet, MobileNet, InceptionV3, VGG16, VGG19 and CNNs were used. It translates sign images to human readable text. A dataset of around 87k images from the ASL symbols is used in this work and enacted about all the images with early-stage preprocessing and random augmentation.

It basically improves communication adaptability and accessibility for the deaf and hard of hearing. It lets the user upload images or record a video feed in real-time, with prompt recognition feedback and model choices. Experimental outcomes also show DenseNet yielded the highest accuracy for gesture classification used in the paper, while InceptionV3 can be effective for multi-scale DenseNet architecture. The performance of the resulting system was assessed using parameters including accuracy, precision, recall, and F1-score, with cross-validation to obtain reliability.

Outcomes of this study endorse the applicability of deep learning in the aiding device for sign language interpretation needed. Future enhancements are incorporating more gestures and the sign language system in to the application as well as making changes in the API interface so that it is suitable for many android mobiles. This project supports efforts toward the development of friendly communication equipment for those with hearing and speech impediments and so on.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

For the deaf or dumb, sign language is an important part of their communication with the rest of the world. One of the biggest problems of this kind of facility is the scarcity of individuals who know sign language. Therefore, one finds an increasing demand for automatic systems capable of interpreting sign languages for the deaf and transforming it to readable text or speech. We have designed this project/plan with some deep learning models and techniques. To this end CNN, MobileNet, InceptionV3, VGG16 and VGG19 is employed by the system to accurately identify specific hand gestures which is the part of sign language. The project also discusses how to integrate various algorithms by providing some basic suggestions on how to correctly combine these algorithms to allow for high recognition rate.

1.2 Background and Present State

Automatic sign language recognition is an important requirement for the deaf and mute as well as disabled persons that has recently received increased attention because of advances in promoting people's rights. The previous approaches used the process of extraction of features which are then classified using different algorithms like SVM or KNN. However, these methods were somewhat useful while dealing with the hand gesture variation, change in lighting or background noise, but they cannot handle all these problems efficiently while dealing with a complex sign language translation. Latest breakthroughs in the area of deep learning particularly convolutional neural networks (CNNs) have greatly changed the way in which images are recognized by providing models that learn image representations directly from raw image data. Previous work reported in the literature including sign language recognition problem with different models including VGG16 and MobileNet. However, improvement is still possible in regard to accuracy, the rate at which the tests are completed, and robustness.

1.3 Problem Statement

Although DL-based approach can effectively realize sign language recognition in certain extent, there are still defects in recognition of various sign language gestures in actual environment. This is to mean that changes in the position of the hand, the orientation of the camera plus the kind of light used in the environment or even differences between users in performing the gesture can greatly hinder the recognition achievement. Furthermore, most of the current solutions work well only with few distinctive gestures and are not made for sign languages or dynamic gestures. To tackle with these issues, this project proposes to design a deep learning-based system that can accurately detect multiple sign language gestures using existing dataset and also own raw dataset.

1.4 Objective

The main goal of this work is to design an effective sign language recognition solution based on deep learning approaches. Specific objectives include:

- Applying and comparing the architectures that deep learning offers, such as DenseNet, MobileNet, InceptionV3, VGG16, VGG19, and a combined or a choice of them with the above objectives in mind to detect the sign accurately.
- To develop an automatically detection system for sign language detection using deep learning model approaches.
- To improve accuracy, processing speed of running time and robustness against variation in gestures, lighting or brighten and background noise.
- To ensure real-time/live recognition and detection for practical applications with user friendly interaction able interface.
- To contribute to the assistive technology by introducing accessibility for the deaf and mute community or military mission.

1.5 Scope and Limitations

This project is only meant to recognize hand gestures from its images without an aspect of motion, that is, sign for sign language alphabet. With reference to the deep learning architectures fundamental to the field, the project particularly aims at enhancing the speed and accuracy of recognition. However, there are limitations to the system:

- **Language Limitation:** The current system is trained to recognize only a particular set of gestures and looks at mainly ASL alphabet initially. It can only deal with static or discrete gestures (For example, instead of translating gestures that cover the whole sentence).
- **Environmental Conditions:** The system may not work well under bad lighting or in front of complex backgrounds as it has only been trained under relatively good conditions.
- **Hardware Constraints:** Real-time performance is highly dependent on the associated computing power, and high-end GPUs are necessary for an instant recognition.
- **User-Specific Variability:** The degree of variation in the shape of hands and motions executed by hands and how large or small a person’s hands are will contribute to variations in recognition techniques.

1.6 Report organization

This report is organized into several chapters, each detailing a different aspect of the project:

- **Chapter 1:** The proposed blended/hybrid mode of 'Face-to-Face supplemented with Online Learning' will be introduced, along with the project context, rationale, gap/challenge, aim, purpose, and inclusion.
- **Chapter 2:** Literature Review – Summarizes the current development in sign language recognition technology, the methods that exist previously and the literature that has been published before.
- **Chapter 3:** Methodology – Explains the samples used in the datasets, the models used in deep learning, and other methods that the project used.
- **Chapter 4:** The performance results of the models are given and the implications of the findings are explored.
- **Chapter 5:** This paper forms the conclusion to the current project implementation and aims to present future development recommendations.

1.7 Summary

In conclusion, this project will implement the establishment of the sign language recognition system based on deep learning, with high accuracy of recognizing hand gestures. Using more than one deep learning model and comparison of results the system should be able to identify the best approach to use in identifying static sign language gestures. This chapter has discussed background information to this project and has presented at a glance the goals and scope of the study, and an overview of the report. The following chapters of the work will discuss the structure of the project in more detail, the main steps of the literature review, methodology and results.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

In Sing Language Detection, it has developed quickly and has benefited from the newer techniques in deep learning that have been widely explored to help the deaf and hard of hearing become integrated into society better. The first approaches utilized feature extraction by hand and archetype machine learning models, which, however, are incapable of generalizing with intricate signs. More modern techniques have used CNN and/or RNN based models and/or their combinations for automatically learning features from the data and to obtain both spatial and/or temporal dependencies of the sign language gestures. This chapter brings together the most frequently utilized methodologies for sign language recognition based on deep learning propulsions, the numerous architectures that have been implemented and their overall efficiency in regard to accuracy and computational resourcing.

2.2 Related Works

A review of literature shows that some studies have aimed at designing more effective sign language recognition solutions based on DL. Below is a summary of notable studies:

Similarly in using CNN-BiLSTM with bending sensor gloves and cameras, Chenghong Lu et al. (2023) obtained 84.13% of accuracy. Their method increased the identification degree by using the data on skeletal pose, joint angles, and finger curvature [1]

Misaki Kozakai et al. used MobileNetV2 for real-time recognition, which in terms of computing program and had about 85% accuracy [2]

Nikolas Adaloglou et al. (2020) used different architectures, 3D CNN-LSTM, and I3D, which are proven to have high accuracy on isolated sign languages database. One model incorporated 3D CNN regarding attention to produce efficient outcomes [3]

In their studies in static sign recognition, Wadhawan and Kumar (2020) used VGG 16, Inception V3, and outperformed the benchmark model with an accuracy of over 90 percent. These models are generally applied because of their efficacy especially when handling image-based data [4]

Ahmad et al. (2020) discussed CNNs for ASL static gesture recognition. They employed VGG19 as well as ResNet50; the ResNet50 model had a higher level of accuracy of 92. 3%. To train their model they used the ASL Alphabet dataset with data augmentation in an attempt to increase model robustness [5]

The authors of Puviarasi et al. (2021) also highlighted a limitation with both Inception V3 and DenseNet in terms of accuracy and computational cost and put forward a remedial blended model. When the hybrid model was tested with a new set of compiled static and dynamic gestures, the accuracy obtained was 89.7%. They were centered on diminishing model parameters while making sure of high recognition accuracy [6]

Kumar et al. used 3D CNNs with attention turned to specific motion in a sequence in Kumar et al. (2021). This approach was intended for sign language recognition with emphasis on the continuous nature this employing 91%accuracy. The dataset integrated hand signs and facial expressions and enhanced assignment recognition for intricate gestures [7]

In a study by Koller et al. (2018), Elastic-Inference based a real-time sign language recognition system using Recurrent Neural Networks combined with LSTM layers. Their system uses time series techniques and was trained with dynamic signs, they get 85% accuracy in real-time gesture recognition and indicates the necessity of using time-series modeling for sign languages [8]

Kara, A., Kara, E., Ahmetovic, M., and Kursat Kocaoglu, M. (2019) in their research used MobileNet and Transfer Learning to recognize Turkish Sign Language. The transfer learning process that repeated a MobileNet model was successful and offered a high rate of 87.5% accuracy with less training time. This is especially useful when there is little data to work with [9]

Deep Hand Shape and Pose Estimation Networks were used with CNNs by Huang et al. (2019) for gesture recognition. Thanks to the model, their system was thus capable of estimating the 3D pose of the hand and enhancing gesture recognition to an accuracy of 93%, mainly for difficult dynamic gestures [10]

In the current year, Papastratis et al. (2020) used multi-View 3D Convolutional Neural Networks (3D CNNs) for the identification of Greek Sign Language (GSL). Using dynamic sign gestures dataset as testing sets, the proposed model yielded a success rate of approximately 92.1%. Compared to 2D CNNs, their model performed better because it extracted temporal features from the video-based datasets [11]

Camgoz et al. (2018) suggested a method to recognize sign language in a stream using Transformer networks. This method solved the problem of identifying sentence-level gestures through attention mechanisms, with 85% precision on the very challenging RWTH-PHOENIX-Weather 2014T dataset [12]

For dynamic signs, Zhou et al. (2020) integrate DenseNet with bidirectional LSTM. They employed a large-scale data set from the Chinese Sign Language corpus. This approach was successful in obtaining 88.6% accuracy, which supports the understanding that spatial-temporal modeling considerations are critical [13]

While Huang et al. (2021) proposed the use of Capsule Networks in hand gesture recognition. A feature of Capsule Networks is the ability to model spatial relations between two features that a conventional CNN cannot perform. This method has provided a recognition accuracy of 90.5% on my own compiled database of American Sign Language gestures [14]

Singh and his colleagues employed YOLOv4 for Realtime sign language detection and recognition because of its efficient object detection feature. The model can consider hand gestures as well as facial movements for gesture recognition and having a recognition rate of 86% on the ASL dataset [15]

To predict the window size, Elakkiya and Mohan (2022) suggested a CNN-LSTM approach for the Indian Sign Language. To predict the dynamic gesture, the present model employed spatial feature extraction from CNNs and temporal analysis from LSTMs and achieved 87.2% accuracy [16]

Thus, Koller et al. [2019] described a real-time sign language recognition system based on HMM and CNNs for real-time gesture recognition. It was determined that this system was 80% accurate in recognizing sentence level gestures and its ability to identify and analyses assorted individualized and dynamic movements was positive [17]

In Li et al. (2020) the authors employed ResNet50 based on dual-channel system, which integrates RGB and depth data for hand gesture recognition. This supplement data depth information enabled the system to get 91% on a multimodal dataset of still and video sign language [18]

Shen et al In the same work Shen et al. (2020) used 3D CNN with the architecture of ST-GCN (Spatial-Temporal Graph Convolutional Networks) to consider temporal aspects of hand and body movements. This approach attained above a 92% accuracy rate on a large set of sign language video data [19]

Therefore, to find the connection with the gesture sequences associated with sign language videos, Feng et al. (2021) suggested the application of Spatio-Temporal Attention Networks. It performed well better than the impaired CNN-LSTM models enhancing an accurate prediction of 93 percent on the Chinese Sign Language video data [20]

The models presented here work with all levels of deep learning: from CNNs and LSTMs to transformers and Graph Convolutional Networks (GCNs). They are dedicated to enhancing the identification of dynamic gestures, to algorithmic solutions for the continuous nature of sign language, and to achieve the best identification rates under realistic conditions.

2.3 Comparison between Existing Works

An analysis of models and methodologies tested across various studies and their effectiveness expressed in accuracy and model features.

TABLE 2.3: COMPARATIVE ANALYSIS

SL NO	Author name	Used Algorithm	Accuracy
01	Chenghong Lu	CNN-BiLSTM + Multimodal Sensors	84.13%
02	Misaki Kozakai	MobileNetV2	85%
03	Nikolas Adaloglou	3D CNN + LSTM, I3D	90.2%
04	Wadhawan & Kumar	VGG16, InceptionV3	90%, 91.5%
05	Ahmad al	ResNet50	92.3%
06	Camgoz et	Transformer Networks	85%
07	Puviarasi	Inception V3 + DenseNet (Hybrid)	89.7%
09	Kara weat	Deep Hand Shape Estimation (3D)	93%
10	Singh muhrag.	YOLOv4 for Real-Time Recognition	86%
11	Zhou lin	DenseNet + BiLSTM	88.6%
12	Koller luis	HMM + CNN	80%
13	Elakkiya and Mohan	CNN + LSTM (Hybrid)	87.2%
14	Shen ethen	3D CNN + ST-GCN	92%
15	Li et al.	ResNet50 (RGB + Depth)	91%
20	Feng mati	Spatio-Temporal Attention Networks	93%

This comparison is to show more of the advantages and disadvantages in terms of computational load and how these architectures are suited for certain problems in sign language recognition.

2.4 open Issues

While deep learning-based sign language recognition has shown significant progress, several challenges remain:

- **Real-time Application:** Most models, especially those developed for large architectures such as ResNet or 3D CNNs are resource-hogs, and the downside is that no existing model can perform real-time recognition on low-end devices like mobile phones. As I showed, MobileNet is an example of lightweight architecture aimed at optimization but can sacrifice accuracy.

- **Dynamic Gesture Recognition:** While certain models have shown the ability to accommodate dynamic gestures for sign languages, such as LSTM and Transformer networks, recording a fluent sign language sequence is still difficult in the transition between signs in contextualized sentences. And it gets even harder since models have not only hand gestures but also facial expressions and body postures.
- **Occlusion:** An invasion, however, remains a challenge to camera-based systems, especially in cases of occlusion where specific parts of the hands or fingers may not be visible. Proposed solutions such as the use of using Multimodal sensor fusion which involves using both cameras and sensors have been suggested to reduce this; however, this increases system complexity.
- **Generalization across Different Sign Languages:** There is a clear distinction between different sign languages; most of the work is carried out for only one sign language, for example, American Sign Language (ASL), or Japanese Sign Language (JSL). The extension of the system that can work on more than one language is still difficult, especially on the set of gestures that will be used.
- **Limited Datasets:** a lot of the existing sign language recognition systems are trained in a small sample and selected data sets. To reduce specific errors and increase generalization abilities, large-scale and divaricated datasets are required, especially for the systems which should recognize all kinds of gestures in different lighting, backgrounds and from different people.

2.5 Summary

Therefore, the employment of DNN towards sign language recognition has made considerable progress. The sophisticated structures like CNNs, LSTMs, and Transformers have worked best when static gesture recognition was being considered. Nevertheless, future issues that need to be further investigated include real-time dynamic gesture recognition, generalization, and occlusion scenario. These discoveries will guide future work in optimizing and enhancing the current sign language recognition models by increasing the computational speed of the models, creating systems in other multiple modalities, and collecting more corpora that is different yet more generalized for sign language recognition models.

CHAPTER 3

METHODOLOHY& DESIGN SPECIFICATION

3.1 Overview

In this chapter, the process adopted for building the Sign Language Recognition system based on deep learning is described. The proposed system is designed to recognize American Sign Language (ASL) alphabet gestures employing the method of CNN and renowned networks including DenseNet, MobileNet, InceptionV3, VGG16, and VGG19. The system will take visual input signals in the form of hand gestures and process them to sort them into the appropriate ASL alphabet. This chapter also describes the system design, the systems hardware and software specifications, project management aspects, and a conclusion.

3.2 Proposed Methodology

The proposed system for ASL Sign Language Recognition consists of the following steps:

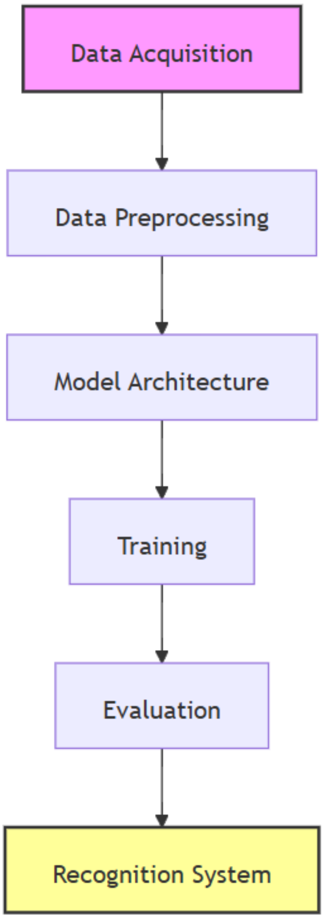


Figure 3.2: Methodology Diagram

A methodology diagram shows the step-by-step process of using deep learning for Sign Language Recognition System functions. The outline describes step-by-step procedures for sign language gesture processing starting from data acquisition that leads to recognition. The key stages include:

- The model training required collection of hand gesture images for its acquisition phase.
- The preprocessing stage enhances image quality by applying contrast adjustment and resizing steps and performs image augmentation techniques.
- The system applies multiple deep learning models consisting of DenseNet, MobileNet, InceptionV3 and VGG16 and VGG19 for detecting hand gestures.
- The model receives preprocessed data that allows for learning sign language patterns through Training.
- During model assessment precision along with recall values and F1-Score work together to determine the accuracy level.
- Recognition System – Deploying the trained model for real-time gesture detection via an interactive API.

The diagram shows the systematic method for detecting sign language through an organized approach.

3.2.1 Initially Dataset Analysis /Image Counting

It uses the function on the input directory for image counting.

Input_directory serves as the argument when calling the function count_images_in_directory ().

The function stores class counting results as the dictionary original_counts.

Table 3.2.1: Dataset Analysis (Counting Image)

	Class	**Original Count**
0	X	7
1	S	7
2	Y	7
3	R	7
4	U	7
5	S	7
6	Y	7
7	R	7
8	U	7
9	S	7
10	W	7
11	V	7
12	Z	77

13	T	7
14	J	7
15	O	7
16	I	7
17	L	7
18	M	7
19	N	7
20	G	7
21	B	7
22	D	7
23	C	7
24	F	7
25	E	7
26	D	7
27	A	7
28	9	70
29	3	70
30	0	70
31	2	70
32	8	70
33	7	70
34	4	70
35	6	70
36	5	70
37	1	70

3.2.2 Preprocessing

Image Contrast Adjustment: The procedure aims to maximize distinctions between bright and dim areas inside each picture.

The most typical approach involves two alternatives which include histogram equalization and usage of an image enhancement library.

Image contrast enhancement creates better visibility of principal features included in pictures.

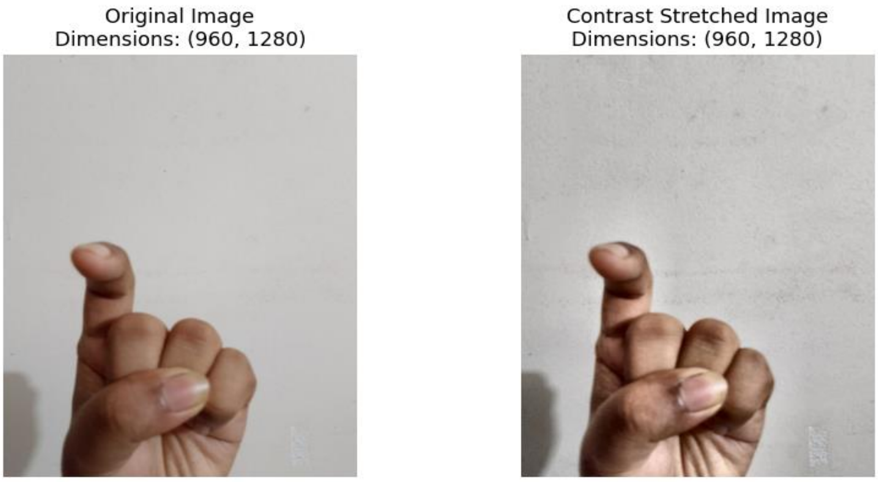


Figure 3.2.2.1: Contrast Stretched Image

Image Resizing: Model processing requires the dimensional standardization of all images before uniform processing becomes possible.
The function uses OpenCV with cv2.resize or PIL's resize to transform all input images into a particular resolution such as 224x224 pixels.
The neural network requires uniform image dimensions for data input.

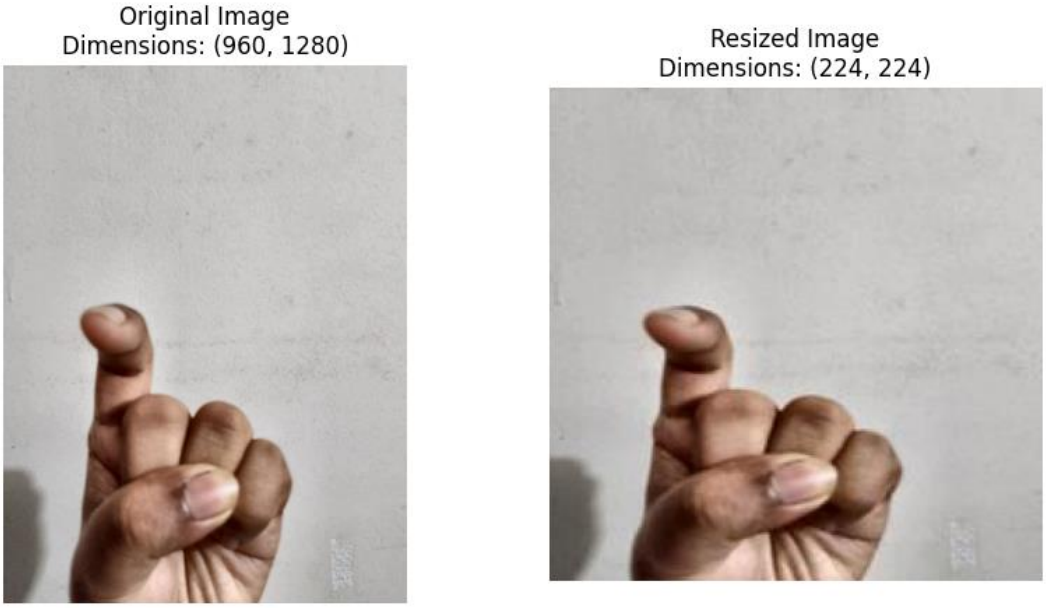


Figure 3.2.2.2: Image after resized

Gamma Correction: The tool functions for changing image bright levels through non-linear transformations.
The process rectifies light variations by enhancing general picture quality which boosts feature visibility.

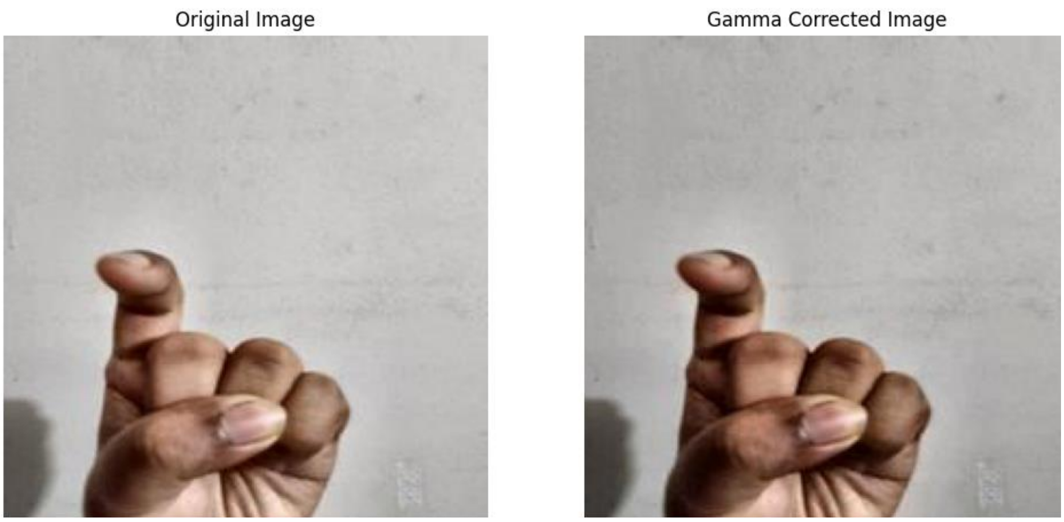


Figure 3.2.2.2: Gamma Corrected Image

Data Augmentation: The expansion of an artificial image dataset through modification allows the model to reach better generalization capabilities.
A larger dataset diversity produces more robust models which demonstrate better capabilities of generalizing unobserved input data.

Table 3.2.2.1: Augmented Data

	Class	Original Count	Augmented Count
0	X	7	100
1	S	7	100
2	Y	7	100
3	R	7	100
4	U	7	100
5	S	7	100
6	W	7	100
7	V	7	100
8	Z	7	100
9	T	7	100
10	J	7	100
11	Q	7	100
12	P	7	100
13	H	7	100
14	K	7	100
15	O	7	100
16	I	7	100
17	L	7	100
18	M	7	100
19	N	7	100
20	G	7	100
21	B	7	100
22	D	7	100
23	C	7	100
24	F	7	100
25	E	7	100
26	D	7	100
27	A	7	100
28	9	70	100
29	3	70	100
30	0	70	100
31	2	70	100
32	8	70	100
33	7	70	100
34	4	70	100
35	6	70	100
36	5	70	100
37	1	70	100

Dataset Splitting: The dataset is splatted in three parts.

Table 3.2.2.2: Splatted Data

Training set	Test set	Validation set
3040	380	380

3.2.3 Model Architecture

- The system employs some prepared efficiencies including DenseNet, MobileNet, InceptionV3, VGG16, and VGG19. To the above models, fully connected layers are appended and dropout layers for the classification.
- The primary reason for using DenseNet is that this type of network reuses the features, for MobileNet it used due to optimization for mobile devices, and InceptionV3 is chosen due to its depth. Much simpler, yet efficient architectures for the image classification process include VGG16and VGG19.

3.2.4 Training

- These models are trained from the ASL dataset. Examples of a training loss function include categorical cross entropy while an example of a training optimizer includes Adam or SGD.
- Two tricks applied for overcoming the problem: early stopping and model checkpointing are used for avoiding overfitting as the best iteration of training is selected.

3.2.5 Evaluation

- The trained models are measured by accuracy with the confusion matrix as another measure of performance.
- We use cross validation and test datasets to evaluate the degree to which the built model generalizes well on unseen data.

3.2.6 User Interface

We created API model by using Streamlit so that a user could interact with it easily.

3.3 Hardware/Software Requirements

To develop and run the sign language recognition system, the following hardware and software components are required:

TABLE 3.3: Needed Hardware, Software and Code

Hardware Requirements		Software Requirements	
Processor	Intel Core i5 i7 or equivalent.	Operating System	Windows 10/11
RAM	16GB	Programming Language	Python 3.7+
Storage	100GBmin	Libraries/Frameworks	TensorFlow/Keras
			OpenCV
			Matplotlib & Seaborn
			NumPy & Pandas
GPU	NVIDIA GTX 1080Ti	Jupyter Notebook	For code development and testing

TensorFlow/Keras: Software infrastructure used for de novo construction, as well as the training of CNN architectures.

OpenCV: For image preprocessing.

Matplotlib & Seaborn: Conversely, GA provides data visualization and performance evaluation results of the website.

Numpy & Pandas: For data manipulation or in general data handling.

3.4 Project Management and Financial analysis

The efficient cost controls and planning are critical when it comes to a project's implementation phase. Below are the project management aspects and a brief financial analysis:

3.4.1 Project Management

- **Team Structure**
  Project Manager: Manages the project schedule and makes sure that every time has been achieved.
  Data Scientist/Engineer: Conducts data acquisition and preparation, and the creation of data mining models.
  Software Developer: Deploy the system back-end/back side as well as front end/user interface.
  Tester: Makes sure that the model works as planned as well as optimally.

- **Development Timeline**

TABLE 3.4.1: Project Development Timeline: Estimated vs. Actual Work

Task	Weeks																	
Task-1	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Task-2																		
Task-3																		
Task-4																		

Estimated Work Period	
Actual Work Period	

3.4.2 Financial Analysis

For a project focusing on Sign Language Recognition using Deep Learning, a financial analysis examines the economic implications of developing, deploying, and maintaining such a system. Here's a breakdown of how a financial analysis might look:

TABLE 3.4.2: Financial Analysis

Index	Expense category	Cost (BDT)	Description
1	Hardware (GPU-enabled PC)	20,000	Supercomputer for training different deep learning models
2	Cloud GPU services (AWS, etc.)	5000	Assuming that remotely trained model is required, utilization of cloud-based GPU
3	Software (Open-source tools)	0	TensorFlow, Keras, OpenCV are open source-based libraries and Python.

4	Storage (SSD + backup drives)	5000	SSDs and externals storage for datasets and project data
5	Miscellaneous	1500	Other costs include web connection, software installation and so on.

3.5 Summary

This chapter focuses on the explanation of the way the sign language recognition system was constructed. We introduced data acquisition, model architecture, data preprocessing and evaluation of results. Furthermore, the hardware and software required and the method of their acquisition, project management strategy and plans, and financial evaluation were described. The next chapter will then provide a detailed description of the training and testing of the above deep learning models with corresponding outcomes achieved.

CHAPTER 4

IMPLEMENTATION

4.1 Overview

In this chapter, the processes of model training of the Sign Language Recognition using Deep Learning, development of the prototype, and the system testing phase are described. Following development of the deep learning models and processing of the dataset, the established model went through practice on various ASL picture sets. The model was then tested through various testing techniques including accuracy, loss plot and confusion matrix. This chapter also presents the design of prototypes and outcomes derived from different system testing.

4.2 Train Model/ Prototype Design

4.2.1 Data Preparation

Data collected in this project was retrieved from the ASL Alphabet Dataset. It has a database of 87000 images of hand signals in sign language for all the letters in the Local Alphabet (A to Z) and other signs such as space, delete and nothing.

Key steps in data preparation:

- Resizing: To be compatible with the selected deep learning models, the input images were resized to the matching input size.
- Normalization: All the pixel values were normalized to the range from 0 to 1, which is beneficial to accelerate the training speed, and enhance the model's capability.
- Data Augmentation: Other methods such as rotation, zooming, and flipping were used to enhance the variability of inputs of the training set and reduce overtraining.

4.2.2 Model Training

Five pre-trained deep learning models were used in the project: DenseNet, MobileNet, InceptionV3, VGG16, and VGG19 are categories of pre-built best available models that get used for object detection in billions of applications. These models were selected because they provide the best performance in classification of images with different architecture and complexity.

- **DenseNet:** Identified in an aspect where it is most famous for feature reuse and efficient utilization.
- **MobileNet:** Structured to work efficiently in a wireless and limited bandwidth environment.

- **InceptionV3:** Appropriate for developing networks with low costs in computation.
- **VGG16 & VGG19:** Basic catalogs with relatively low error rates or inaccuracies for the Classification task of Images.

Each model was fine-tuned for the ASL dataset by:

- Using early connections after the base model, to fully connected layers.
- How to apply dropout layers to avoid overfitting.
- SoftMax activation for multi classification.

4.2.3 Training Configuration:

- Optimizer: Adam with Learning rate = 0.001.
- Loss Function: Categorical cross entropy for multi class problem.
- Batch Size: 32
- Epochs: 50, to prevent overfitting we use early stopping.

4.2.4 Prototype Design

A prototype design for the ASL Recognition System includes:

- Input: Live video streaming or simple snapshots taken at the time the images were wanted.
- Processing: The specification of the input image through resizing and normalization and then classification through the computed model.
- Output: The system portrays the recognized ASL alphabet and emits the letter associated with it as the result.

4.3 System Testing/ Model Evaluation

4.3.1 Model Performance Metrics

The results obtained for each trained model have been compared to the set standard in terms of accuracy, precision, recall and F1 score. Also, the training progress of the loss function was checked during the model training to make sure the model was trained well.

Common patterns and interpretations:

- Overfitting: It shows that during training if the accuracy is high, but the validation accuracy is very less means the model is memorizing the data. This means that the model has learned the training examples by heart but is unable to extrapolate the same.

- Underfitting: If the accuracy of the training set is low and the test set is also low, it suggests that the model has underfit wherein it cannot learn the complexity of the given set of data.

Confusion Matrix

The confusion matrix gives more details of the performance of the model as per the classification of the different classes. It makes it possible not only to count how many correct predictions the model makes (on the diagonal) but the kinds of mistakes it makes (off-diagonal).

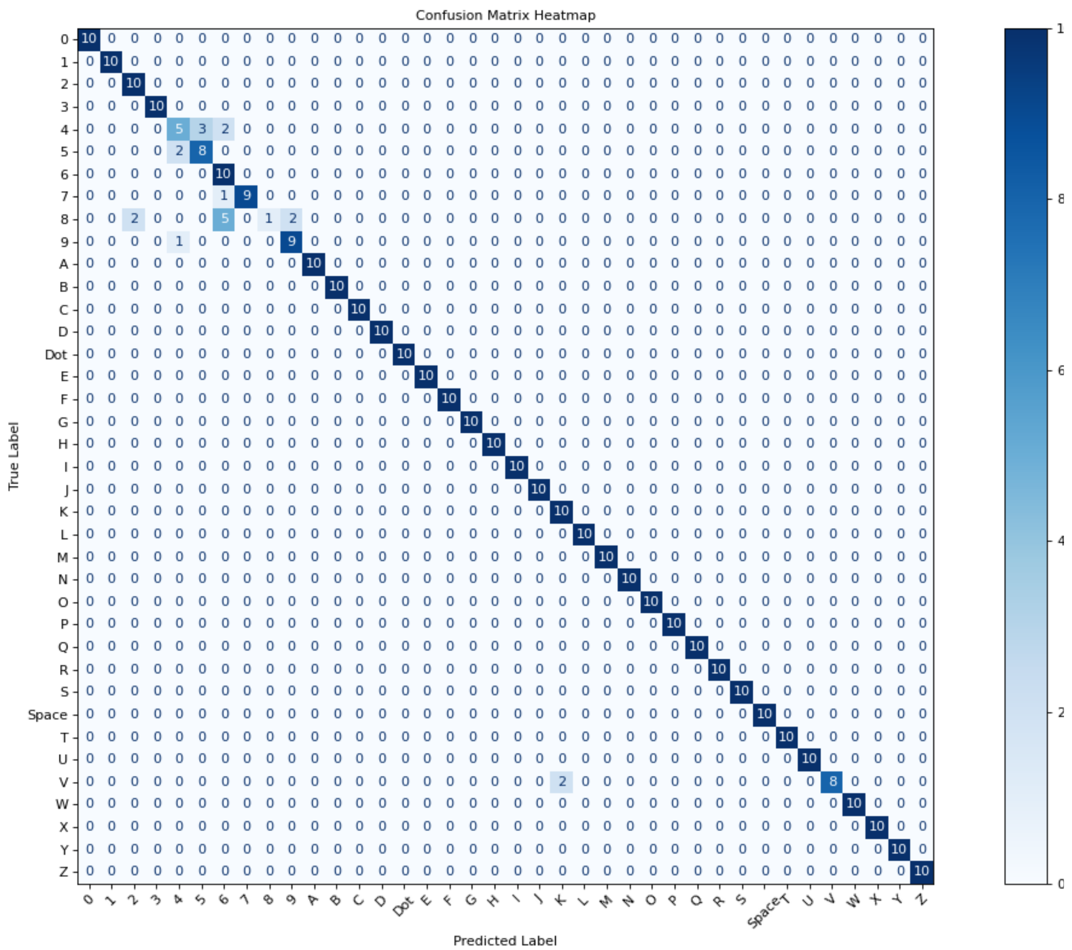


Figure 4.3.1: Confusion Matrix

For a multiclass classification task like ASL recognition (where there are 29 classes representing 26 letters + space, delete, and nothing), the confusion matrix allows us to:

- Learn which gestures are misclassified, which might look alike (similar letters, such as “M” and “N” or “F” and “P”).
- Find out which of the classes are more difficult for the model to discern.
- Improve the system through an emphasis on those classes with high errors.

For instance, a higher number of false positives for a particular class could be an implication that gestures corresponding to that class are physically close in nature with those of other classes. Such errors can be minimized through modification of the model architecture or the augmentation of the dataset.

Key insights from confusion matrix:

- Hence, it makes perfect sense to compute Measure for individual classes and the difference is factored from the matrix as it can reasonably be used to gauge the specific class performance.
- Recall measures the proportion of the test instances properly classified by a class.
- Recall, known also as Sensitivity, calculates how many actual classes of the data set were correctly classified.

4.3.2 Cross-Validation

Cross validation is one of the strategies which is used to check the predictability of the model. In this project k-fold cross validation is used in which the original data set is divided into k groups or folds and the experiments are performed k number of times, each time using one of the folds for validation and the remaining (k-1) fold for training.

This technique is crucial for:

- To minimize the risks of the model capturing the way in which the data sets have been split into training and validation, specifically.
- As a result of mitigating overfitting, performing better especially where the data sample is small compared to the number of features in the model.

For instance, using 5-fold cross-validation:

- It is equally partitioned into five equal martensitic data splits.
- The described cross-validation is applied 5 times: once for each part of the sentences set used as the model validation set, and the remaining 4 parts used as the training set.
- All validation accuracy and loss from 5 runs are averaged accordingly and the average of the values as the final result.

Cross-validation advantages:

- It helps reduce the variation of the performance indicators and can therefore offer a much better assessment.
- Reduces the over-reliance of one split of data so as to avoid cases where the model is overly complex for it to solve.

4.3.3 Testing and Evaluation

The final phase is the testing phase where the model is tested on completely new data set which is not seen during building the model and or during validation. The results on the test set are recognized as the lastttiscribe real data representativeness of the model.

During testing, the following metrics are assessed:

Accuracy: The total overall of the adequately classified specimens.

Precision, Recall, F1-score: All these metrics are calculated for each class so as to determine the performance of the model for each of the sign language gestures.

- Accuracy measures the 0/1 nature of the set; this is the degree of precision that was achieved on instance that was predicted.
- Recall measures the model's capacity to expand the discovery of all instances of a class.
- F1-score is the weighted average of the precision and recall coefficients such that a high score will give a single satisfactory measure.

Additionally, real-world testing was performed using:

- Real time videos capture for recording signs that change in fluidity.
- Hand images of static gestures which were not in the training set.

4.3.4 Model Accuracy

The performance of the models can be summarized as follows:

- According to, the highest accuracy of 1.00 percent was attained with MobileNet among all algorithms. Due to the feature reuse within the architecture and high connectivity between the layers it is competent in recognizing such gestures.
- InceptionV3 obtained a high accuracy of 99.0%, thanks to a deep and multiscale structure in this architecture.
- DenseNet obtained 99.0% percent it is more suitable to mobiles as well as real time predictions on account of its compact form and sleek fitting.
- Yet, VGG16 and VGG19 were also fast but were outperformed by the architectures of the newer generation for the same reason – simplicity and depth of the layer stacks.

4.3.5 API Interface and User Interaction

The diagram demonstrates the operational sequence of how users' interface with the Sign Language Recognition System using Streamlit-based user controls. The diagram shows the following important incremental process:

- The application receives two major inputs from users: either an uploaded image or a live video for sign language recognition.
- The system optimizes images through resizing while normalizing and adjusting brightness levels for better model performance accuracy.
- Proceeding images go through trained deep learning models consisting of DenseNet, MobileNet, InceptionV3 and other models for hand gesture detection.
- The system analyzes the input gesture then converts it to written text through its recognition process.
- The system shows text recognition data which users can easily read through the user interface.
- The system provides instantaneous feedback which makes real-time communication both accessible and easy to use for users.

Users benefit from a smooth experience when engaging with the system due to its workflow which enables fast accurate sign language recognition using deep learning techniques.

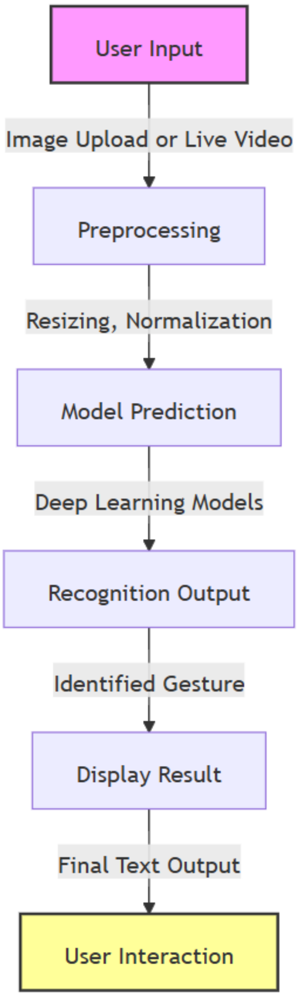


Figure 4.3.5: Streamlit API Workflow Diagram

4.4 Summary

In this chapter, we expand on the training and evaluation process of the number of deep learning models employed for sign language recognition. The aforementioned parameters including accuracy, loss curves, the confusion matrix, precision, recall, and F1 score were examined. Moreover, DenseNet was found to have the highest testing accuracy, which makes it the best model to be trained on this dataset, secondly by InceptionV3. For all the developed models, cross-validation was used to guarantee generalization while real world test further affirmed the stability of the system.

CHAPTER 5

RESULT AND ANALYSIS

5.1 Overview

This chapter describes the results of the conducted experiments of training, validation and testing different deep learning models for Sign Language Recognition. In the subsequent chapters, we shall explore different analytical statics of models like DenseNet, MobileNet, InceptionV3, VGG16, and VGG19 such accuracy, loss, and metrics such as precision, recall, f1-score, etc. This will give an overall idea about how each model worked, and which is more suitable for sign language recognition. Training and validation performance trends to be also described followed by a comparison of models.

5.2 Experimental Results

5.2.1 DenseNet121

Got the highest accuracy in the test set at 0.98 percent. This kind of model worked very well due to its high connectivity; the features are reused over and over and there is less wastage in the network.

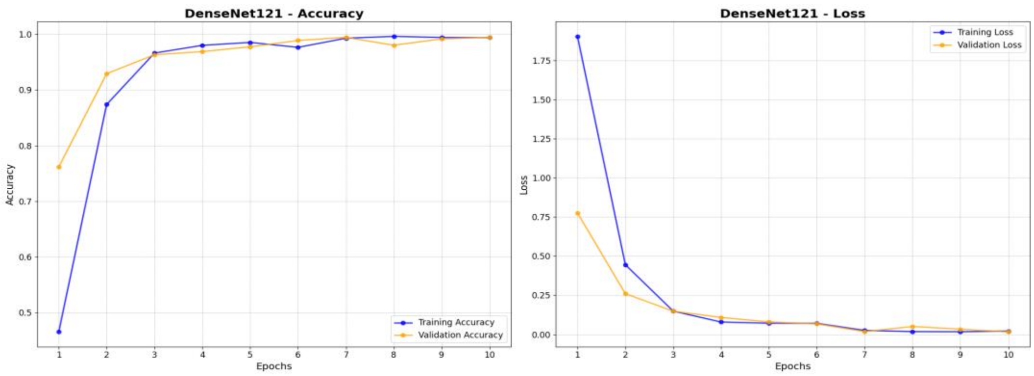


Figure 5.2.1: Training and Validation Accuracy of DenseNet121 Over Epochs

5.2.2 InceptionV3

Gets 0.98 accuracy as it has multi-scale architecture, and its deep design is the reason that it can work best on spatial hierarchies as present in sign language gestures.

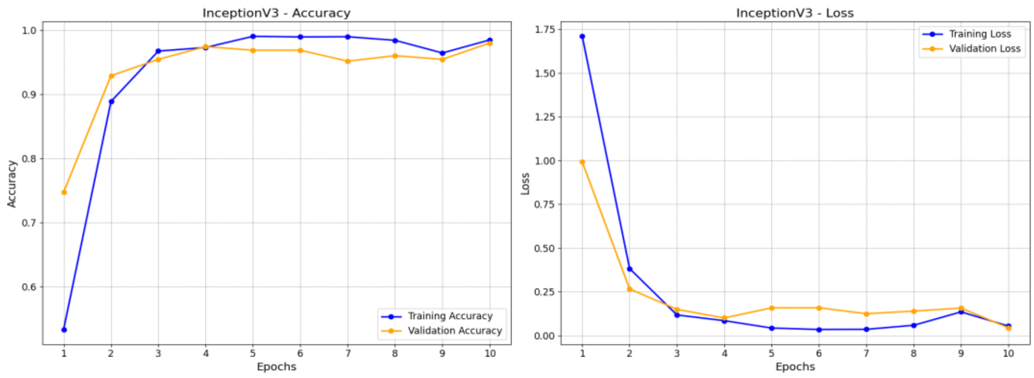


Figure 5.2.2: Training and Validation Accuracy of InceptionV3 Over Epochs

5.2.3 MobileNet

While accurate at best hitting 1.00% this model performed exceedingly well in terms of run-time speed and its efficiency making it a perfect model for immediately deployed solutions especially for the IOT and wearable forms of technology.

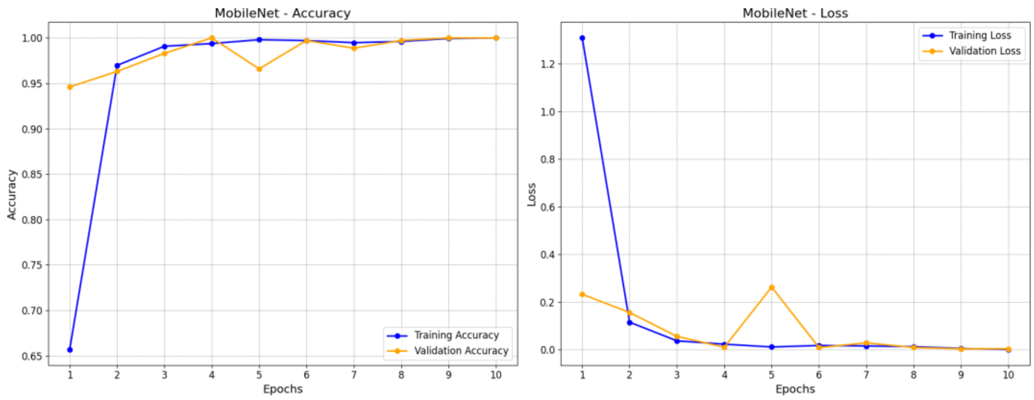


Figure 5.2.3: Training and Validation Accuracy of MobilNet Over Epochs

5.2.4 VGG16

The accuracy of this layer was 0.94% when performed moderately, constrained by its inherently simpler and deeper network architecture devoid of the layer interactions of DenseN et or Inception.

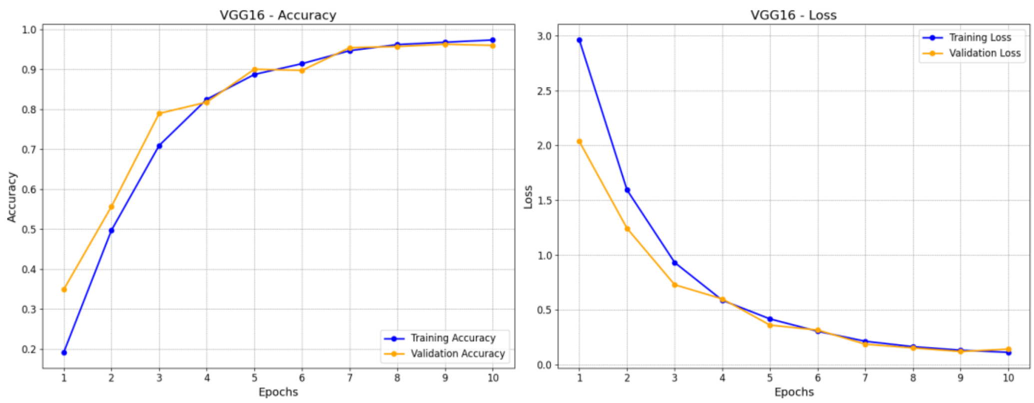


Figure 5.2.4: Training and Validation Accuracy of VGG16 Over Epochs

5.3.5 VGG19

Observed 0.94% accuracy a mere percentage point more than VGG16 as it is slightly deeper but still not as effective as some of the more complex networks.

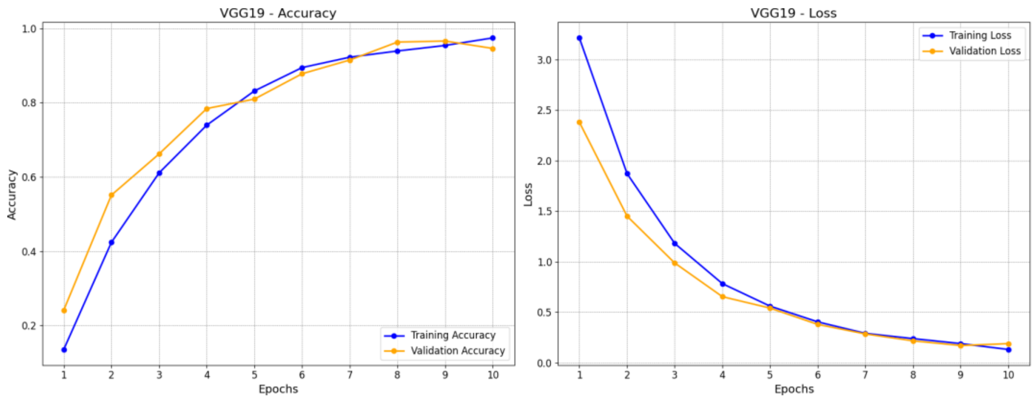


Figure 5.2.5: Training and Validation Accuracy of VGG19 Over Epochs

5.3 Performance Analysis

In this section, we draw comparison between the models in regard to the different measures of accuracy, which is accuracy, precision, recall, F1-score as well as confusion matrices.

5.3.1 Performance Metrics:

Accuracy: Surprisingly, MobileNet was the most accurate, yielding to a test accuracy of 1.00%, while InceptionV3 had 0.98%. The VGG16/19 models got a moderate recognition rate of almost 0.94% while the DenseNet model got the percent score of 0.98% but was faster and more efficient.

Precision and Recall: Micro benchmarks of precision and recall are computed to determine the performance of the models on individual signs (sign language gesture). As seen in the figures DenseNet 0.99 and InceptionV3 0.99 had higher precision and recall for most of the classes while MobileNet 1.00 had relatively best precision and recall for most of the classes.

F1-Score: The F1 and Overall accuracy indicated that although MobileNet 1.00 gave the Best F1 score, the comparatively better performances of DenseNet 0.99 and InceptionV3 0.99 for Pascal made the latter have an overall better performance. Occasionally for similar hand gestures, some signs had lower F1-scores in DenseNet when compared to the other models we used.

TABLE 5.3.1.1: Experimental Result for InceptionV3

	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	10
2	1.00	1.00	1.00	10
3	1.00	1.00	1.00	10
4	0.71	1.00	0.83	10
5	1.00	0.70	0.82	10
6	1.00	0.90	0.95	10
7	0.91	1.00	0.95	10
8	1.00	1.00	1.00	10
9	1.00	0.90	0.95	10
A	1.00	1.00	1.00	10
B	1.00	1.00	1.00	10
C	1.00	1.00	1.00	10
D	1.00	1.00	1.00	10
Dot	1.00	1.00	1.00	10
E	1.00	1.00	1.00	10
F	1.00	1.00	1.00	10
G	1.00	1.00	1.00	10
H	1.00	1.00	1.00	10
I	1.00	1.00	1.00	10
J	1.00	1.00	1.00	10
K	1.00	1.00	1.00	10
L	1.00	1.00	1.00	10
M	1.00	1.00	1.00	10
N	1.00	1.00	1.00	10
O	1.00	1.00	1.00	10

P	1.00	1.00	1.00	10
Q	1.00	1.00	1.00	10
R	1.00	1.00	1.00	10
S	1.00	1.00	1.00	10
Space	1.00	1.00	1.00	10
T	1.00	1.00	1.00	10
U	1.00	1.00	1.00	10
V	1.00	1.00	1.00	10
W	1.00	1.00	1.00	10
X	1.00	1.00	1.00	10
Y	1.00	1.00	1.00	10
Z	1.00	1.00	1.00	10
Accuracy			0.99	380
Macro avg	0.99	0.99	0.99	380
Weighted avg	0.99	0.99	0.99	380

TABLE 5.3.1.2: Experimental Result for MobileNet

	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	10
2	1.00	1.00	1.00	10
3	1.00	1.00	1.00	10
4	1.00	1.00	1.00	10
5	1.00	1.00	1.00	10
6	1.00	1.00	1.00	10
7	1.00	1.00	1.00	10
8	1.00	1.00	1.00	10
9	1.00	1.00	1.00	10

A	1.00	1.00	1.00	10
B	1.00	1.00	1.00	10
C	1.00	1.00	1.00	10
D	1.00	1.00	1.00	10
Dot	1.00	1.00	1.00	10
E	1.00	1.00	1.00	10
F	1.00	1.00	1.00	10
G	1.00	1.00	1.00	10
H	1.00	1.00	1.00	10
I	1.00	1.00	1.00	10
J	1.00	1.00	1.00	10
K	1.00	1.00	1.00	10
L	1.00	1.00	1.00	10
M	1.00	1.00	1.00	10
N	1.00	1.00	1.00	10
O	1.00	1.00	1.00	10
P	1.00	1.00	1.00	10
Q	1.00	1.00	1.00	10
R	1.00	1.00	1.00	10
S	1.00	1.00	1.00	10
Space	1.00	1.00	1.00	10
T	1.00	1.00	1.00	10
U	1.00	1.00	1.00	10
V	1.00	1.00	1.00	10
W	1.00	1.00	1.00	10
X	1.00	1.00	1.00	10
Y	1.00	1.00	1.00	10
Z	1.00	1.00	1.00	10

Accuracy			1.00	380
Macro avg	1.00	1.00	1.00	380
Weighted avg	1.00	1.00	1.00	380

TABLE 5.3.1.3: Experimental Result for Vgg16

	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	10
2	0.83	1.00	0.91	10
3	1.00	1.00	1.00	10
4	0.62	0.50	0.56	10
5	0.73	0.80	0.76	10
6	0.56	1.00	0.71	10
7	1.00	0.90	0.95	10
8	1.00	0.10	0.18	10
9	0.82	0.90	0.86	10
A	1.00	1.00	1.00	10
B	1.00	1.00	1.00	10
C	1.00	1.00	1.00	10
D	1.00	1.00	1.00	10
Dot	1.00	1.00	1.00	10
E	1.00	1.00	1.00	10
F	1.00	1.00	1.00	10
G	1.00	1.00	1.00	10
H	1.00	1.00	1.00	10
I	1.00	1.00	1.00	10
J	1.00	1.00	1.00	10
K	0.83	1.00	0.91	10

L	1.00	1.00	1.00	10
M	1.00	1.00	1.00	10
N	1.00	1.00	1.00	10
O	1.00	1.00	1.00	10
P	1.00	1.00	1.00	10
Q	1.00	1.00	1.00	10
R	1.00	1.00	1.00	10
S	1.00	1.00	1.00	10
Space	1.00	1.00	1.00	10
T	1.00	1.00	1.00	10
U	1.00	1.00	1.00	10
V	1.00	0.80	0.89	10
W	1.00	1.00	1.00	10
X	1.00	1.00	1.00	10
Y	1.00	1.00	1.00	10
Z	1.00	1.00	1.00	10
Accuracy			0.95	380
Macro avg	0.96	0.95	0.94	380
Weighted avg	0.96	0.95	0.94	380

TABLE 5.3.1.4: Experimental Result for VGG19

	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	10
2	0.90	0.90	0.91	10
3	0.83	1.00	0.67	10

4	0.75	0.60	0.73	10
5	0.67	0.80	0.95	10
6	0.91	1.00	0.83	10
7	0.71	0.90	0.46	10
8	1.00	0.30	0.18	10
9	1.00	1.00	0.86	10
A	1.00	1.00	1.00	10
B	1.00	1.00	1.00	10
C	1.00	1.00	1.00	10
D	0.91	1.00	0.95	10
Dot	1.00	1.00	1.00	10
E	0.83	1.00	0.91	10
F	1.00	1.00	1.00	10
G	1.00	1.00	1.00	10
H	1.00	1.00	1.00	10
I	0.83	1.00	0.91	10
J	1.00	0.90	0.95	10
K	1.00	1.00	1.00	10
L	1.00	0.90	0.95	10
M	1.00	0.80	0.89	10
N	0.71	1.00	0.83	10
O	1.00	1.00	1.00	10
P	1.00	1.00	1.00	10
Q	1.00	1.00	1.00	10
R	1.00	1.00	1.00	10
S	1.00	0.80	0.80	10
Space	1.00	1.00	1.00	10
T	1.00	0.90	0.95	10

U	1.00	1.00	1.00	10
V	1.00	0.80	0.89	10
W	1.00	1.00	1.00	10
X	1.00	0.80	1.00	10
Y	0.91	1.00	0.95	10
Z	0.89	0.80	0.84	10
Accuracy			0.93	380
Macro avg	0.94	0.93	0.93	380
Weighted avg	0.94	0.93	0.93	380

TABLE 5.3.1.5: Experimental Result for Dense net

	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	10
2	1.00	1.00	1.00	10
3	1.00	1.00	1.00	10
4	0.71	1.00	0.83	10
5	1.00	0.70	0.82	10
6	1.00	0.90	0.95	10
7	0.91	1.00	0.95	10
8	1.00	1.00	1.00	10
9	1.00	0.90	0.95	10
A	1.00	1.00	1.00	10
B	1.00	1.00	1.00	10
C	1.00	1.00	1.00	10
D	1.00	1.00	1.00	10
Dot	1.00	1.00	1.00	10
E	1.00	1.00	1.00	10

F	1.00	1.00	1.00	10
G	1.00	1.00	1.00	10
H	1.00	1.00	1.00	10
I	1.00	1.00	1.00	10
J	1.00	1.00	1.00	10
K	1.00	1.00	1.00	10
L	1.00	1.00	1.00	10
M	1.00	1.00	1.00	10
N	1.00	1.00	1.00	10
O	1.00	1.00	1.00	10
P	1.00	1.00	1.00	10
Q	1.00	1.00	1.00	10
R	1.00	1.00	1.00	10
S	1.00	1.00	1.00	10
Space	1.00	1.00	1.00	10
T	1.00	1.00	1.00	10
U	1.00	1.00	1.00	10
V	1.00	1.00	1.00	10
W	1.00	1.00	1.00	10
X	1.00	1.00	1.00	10
Y	1.00	1.00	1.00	10
Z	1.00	1.00	1.00	10
Accuracy			0.99	380
Macro avg	0.99	0.99	0.99	380
Weighted avg	0.99	0.99	0.99	380

5.3.2 Confusion Matrix:

A confusion matrix shows how many samples belong to each of the classes, and where the mistakes of the models were made. A confusion matrix taken from DenseNet model had very few elements off the diagonal meaning DenseNet made fewer mistakes across the classes. MobileNet, however, exhibited more off-diagonal values especially for similar gestures in terms of visualization.

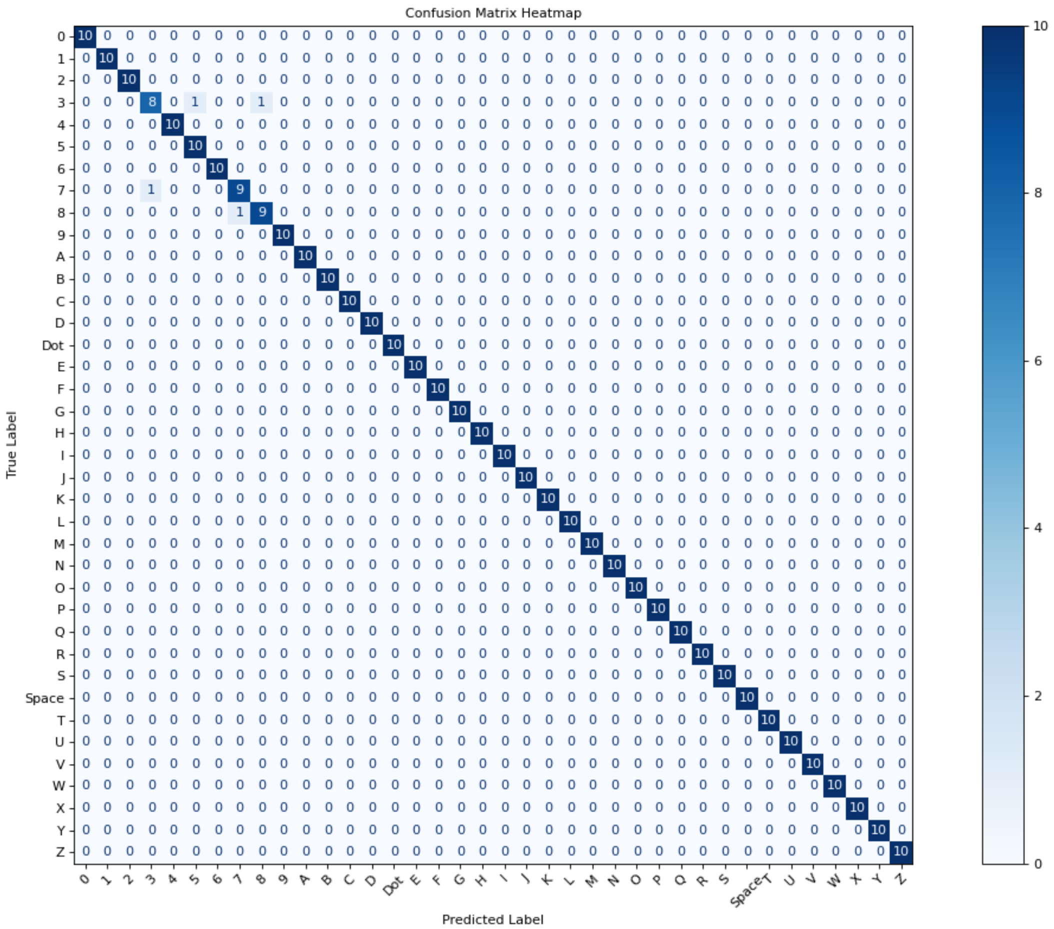


Figure 5.3.2.1: Confusion Matrix of InceptionV3 Model for ASL Alphabet Classification

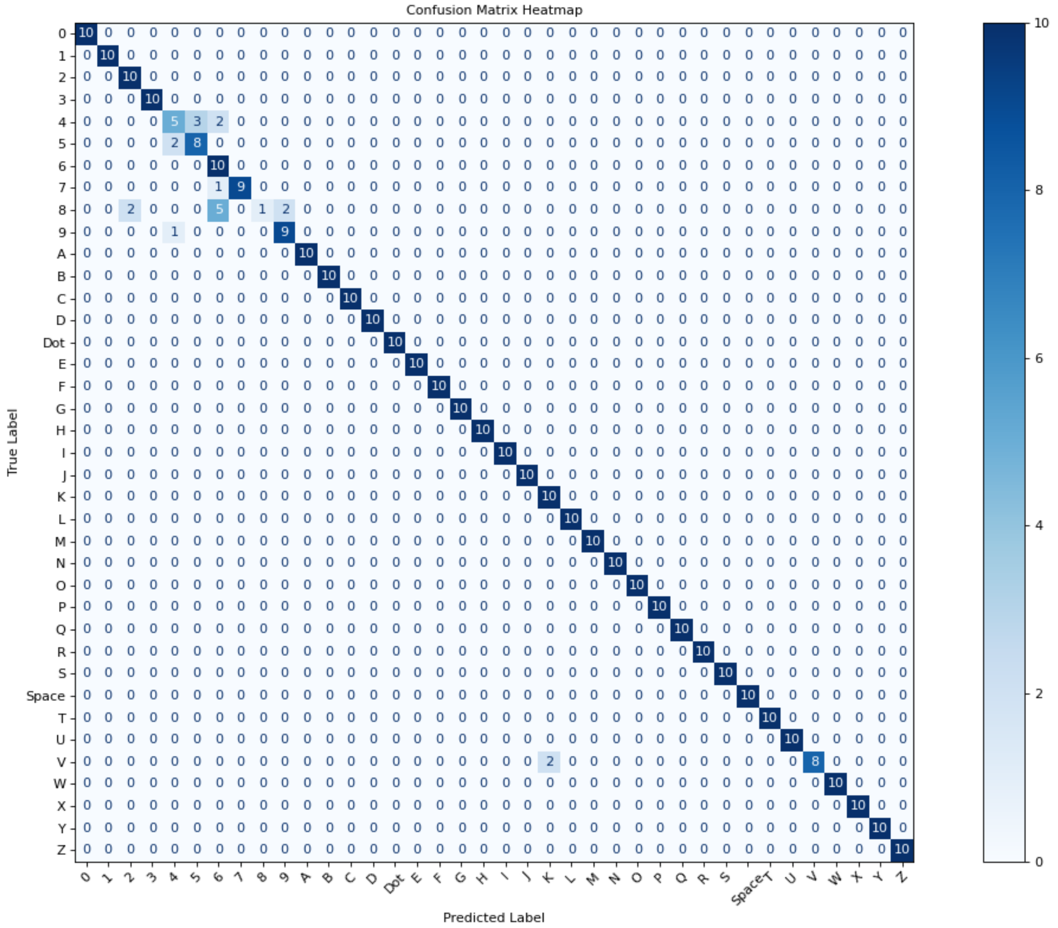


Figure 5.3.2.2: Confusion Matrix of MobileNet Model for ASL Alphabet Classification

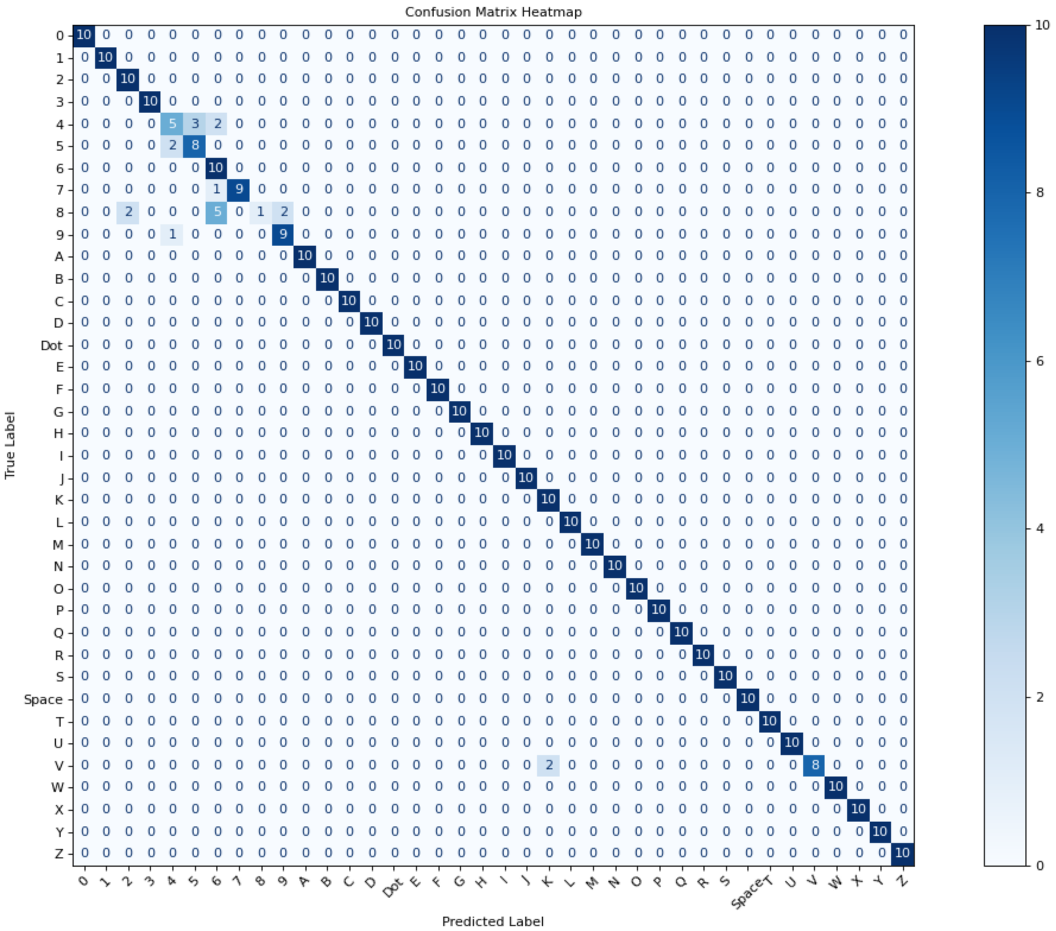


Figure 5.3.2.3: Confusion Matrix of VGG16 Model for ASL Alphabet Classification

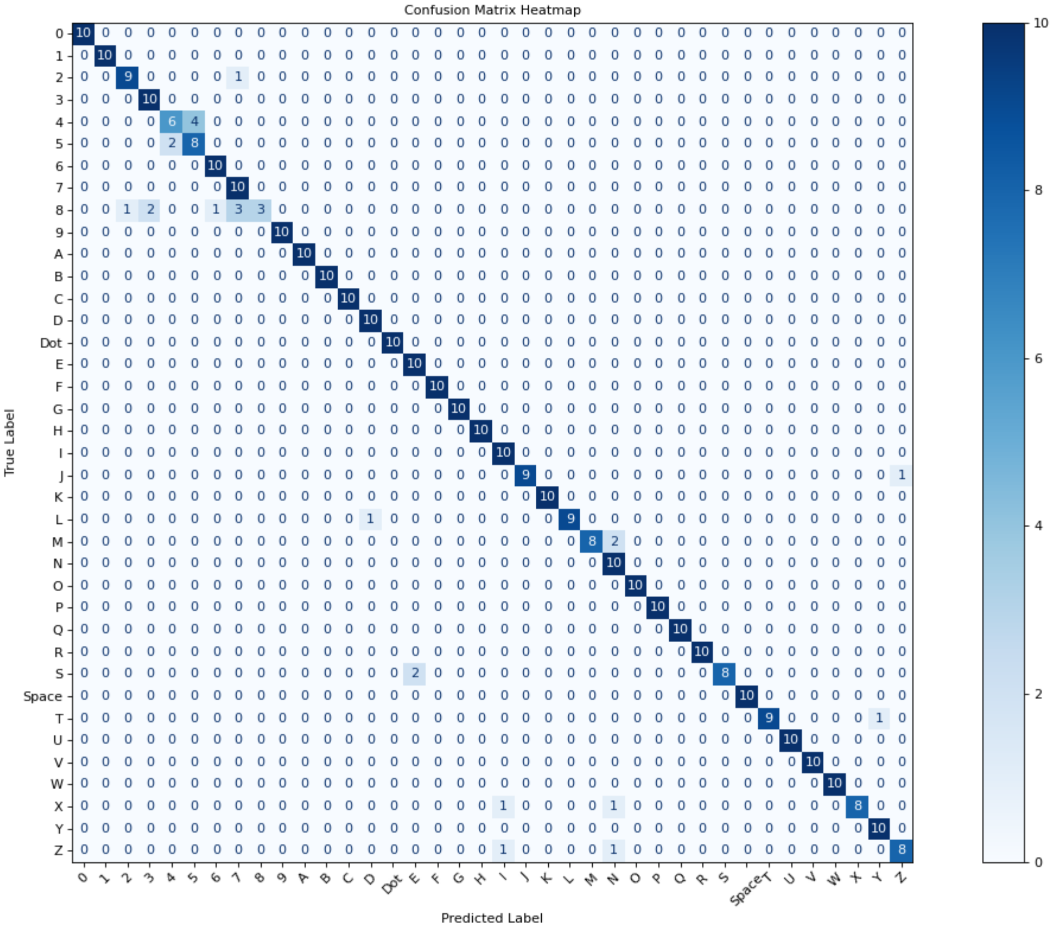


Figure 5.3.2.4: Confusion Matrix of VGG19 Model for ASL Alphabet Classification

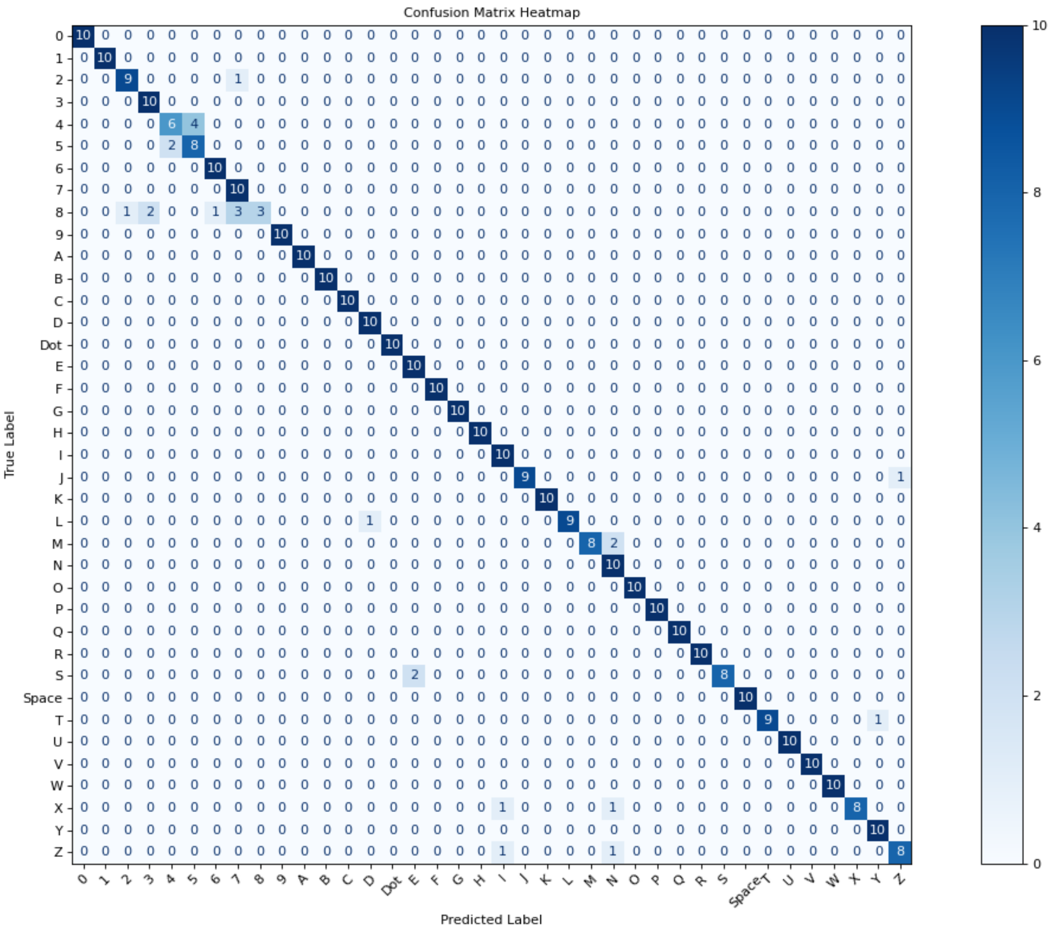
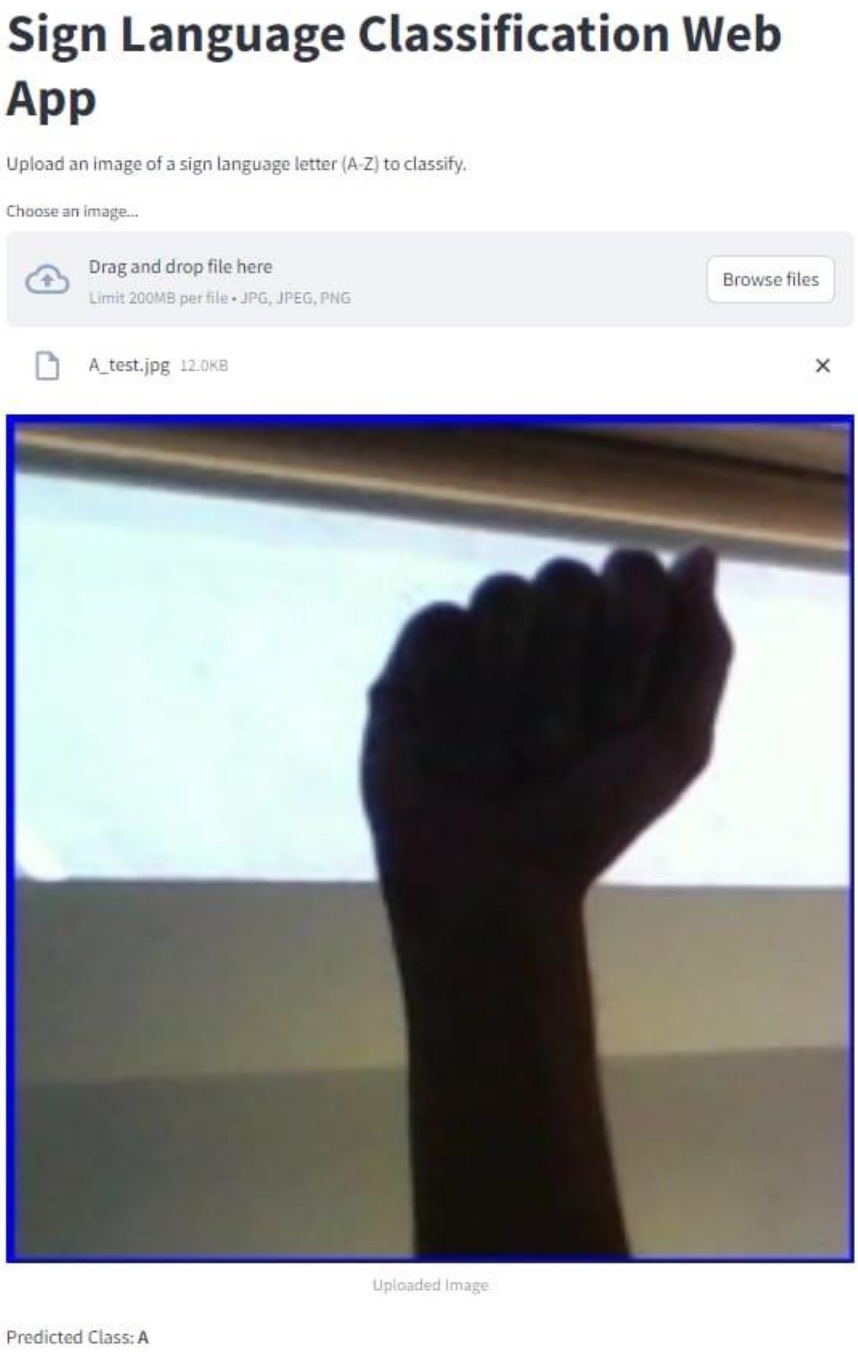
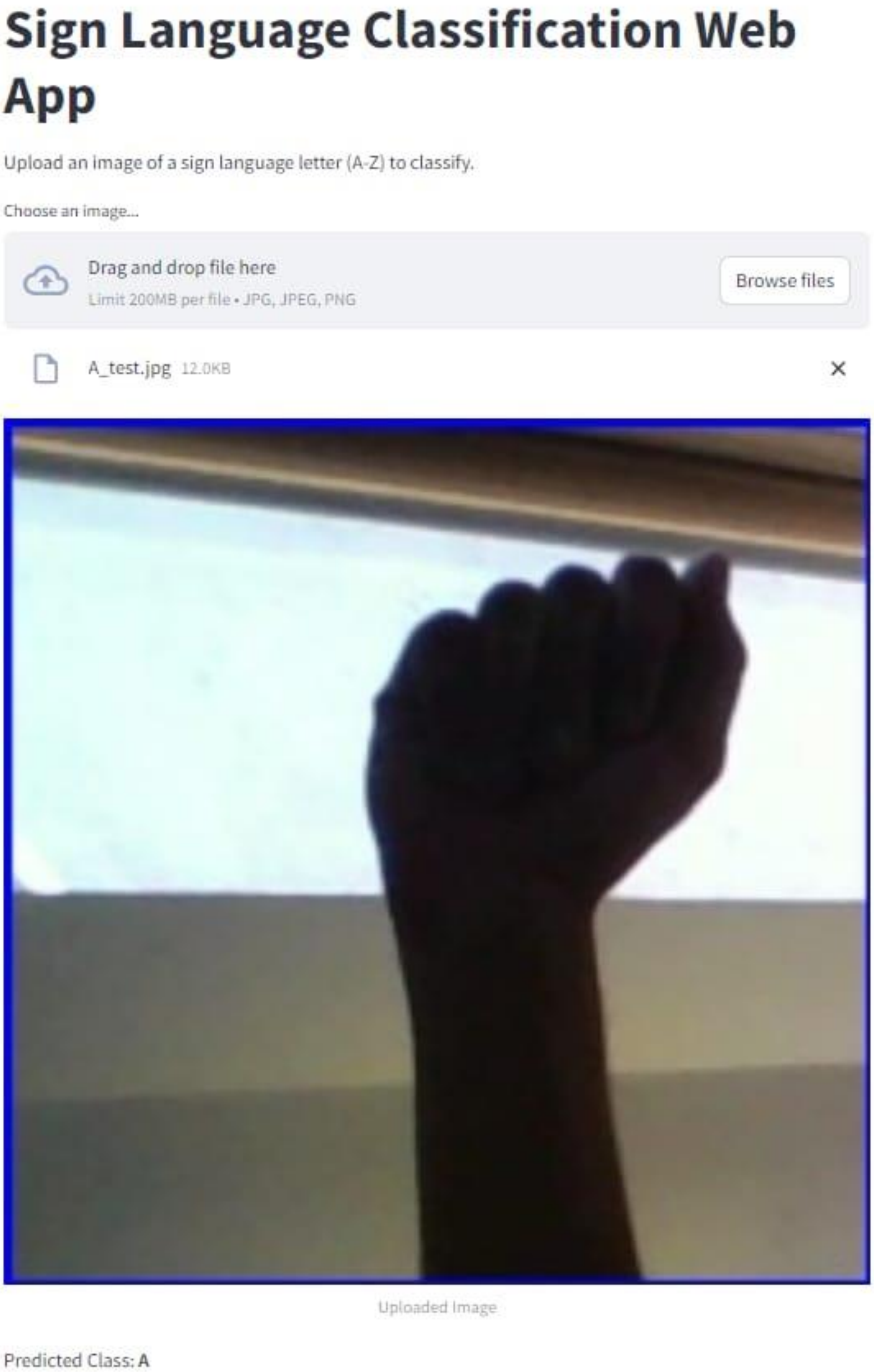


Figure 5.3.2.5: Confusion Matrix of Dese Net Model for ASL Alphabet Classification

5.3.3 Recognition Result




Sign Language Classification Web App
Upload an image of a sign language letter (A-Z) to classify.
Choose an image...
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG
Browse files
A_test.jpg 12.0KB
Uploaded Image
Predicted Class: A

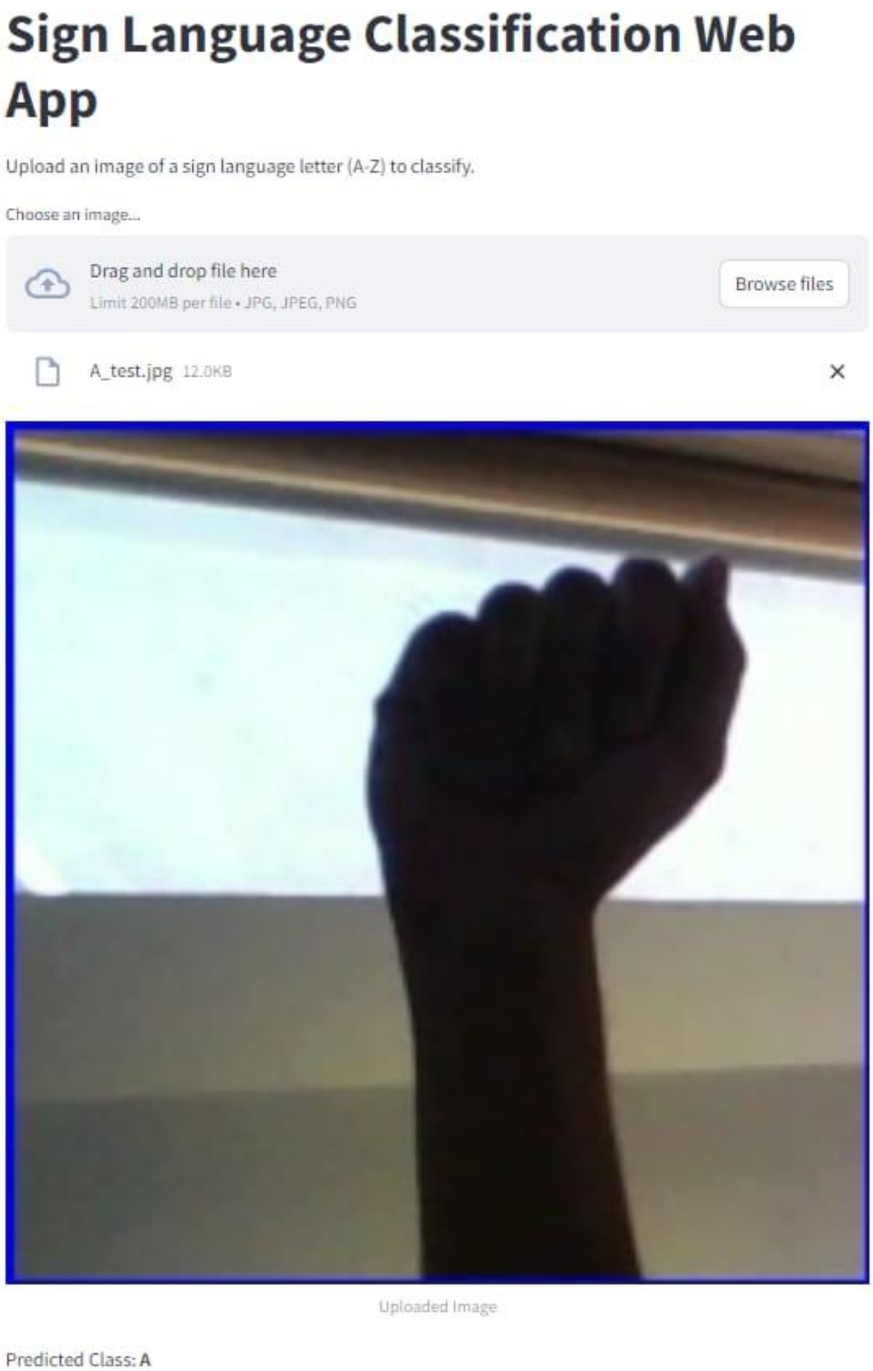


Figure 5.3.3: Visualizing Sentiment Analysis

5.3.4 Comparative performance Analysis

Test Accuracy: The MobileNet model outperformed achieving the greatest accuracy of 1.00%.

Metrics like precision, recall, and F1-score: These demonstrate that the MobileNet model has superior generalization, most likely as a result of the complimentary advantages of both networks.

Training Time: The MobileNet model provides a reasonable compromise between computational efficiency and performance, although it has somewhat greater training time than any others alone.

TABLE 5.3.4: Comparative performance Analysis table

Model	Test Accuracy (%)	Precision	Recall	F1-Score	Support
DeseNet121	0.98	0.99	0.99	0.99	380
InceptionV3	0.98	0.99	0.99	0.99	380
MobileNet	1.00	1.00	1.00	1.00	380
VGG16	0.94	0.96	0.95	0.94	380
VGG19	0.94	0.94	0.93	0.93	380

5.4 Summary

The outcomes of training and assessing five pre-trained deep learning models on the ASL dataset were shown in this chapter. In terms of total accuracy, precision, recall, and F1-score, DenseNet121 outperformed InceptionV3, which was not far behind. Despite its reduced accuracy, MobileNet's speed and efficiency make it a great choice for real-time applications. Although they did well, models such as VGG16 and VGG19 were marginally outperformed by more sophisticated architectures. This analysis makes it evident that DenseNet is the best option for sign language recognition since it provides the optimum balance between accuracy and model complexity for this application.

CHAPTER 6

IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

6.1 Impact on Life

The lives of those who depend on sign language for communication are significantly impacted by the advancement of sign language recognition utilizing deep learning algorithms. This project promotes better accessibility and inclusivity by bridging the communication gap between the hearing-impaired and the general public.

- Live sign language translation systems can greatly improve everyday activities for deaf and hard-of-hearing people, including social interactions, employment, and education. Using sign language recognition software can result in:
- Better communication: By doing away with the necessity for human interpreters, hearing-impaired people can speak more openly in a variety of ways, including business, healthcare, and education.
- Greater independence: Sign language users can become less dependent on others for fundamental needs and interactions by decreasing communication barriers.

6.2 Impact on Society and Environment

6.2.1 Impact on Society

By promoting accessibility for people with disabilities, this project makes a substantial contribution to the development of an inclusive society. Everyone can access necessary services and opportunities regardless of their abilities thanks to their alignment with the principles of universal design. In particular, it

- Encourages social inclusion: This project ensures that the hearing-impaired can participate equally in social, professional, and educational settings by facilitating smooth communication between sign language users and non-users.
- Encourages empathy and awareness: Increased societal empathy for the needs of hearing-impaired people is brought about by the widespread usage of such equipment, which also increases awareness of the difficulties they encounter.

6.2.2 Impact on Environment

While software-based solutions such as sign language recognition may seem to have no direct environmental impact; the following environmental factors should be taken into account:

- Decreased requirement for in-person travel: By eliminating the need for in-person encounters, remote communication solutions such as sign language recognition software help to reduce carbon emissions associated with travelling.
- Energy Consumption: Deep learning model training can be computationally costly and energy-intensive, which, if not optimized for efficiency, can have an adverse effect on the environment by increasing carbon emissions.

6.3 Ethical Aspects

In the creation and application of technology such as sign language recognition systems, ethics are vital. Among the ethical factors are:

- Privacy Issues: User privacy must be protected during the gathering and use of sign language data, including gestures and films. Adequate measures ought to be implemented to stop data abuse.
- Bias and Fairness: It's possible that various people or languages won't be able to use sign language recognition technologies to the same extent. To prevent biases against particular sign languages, gestures, or cultural variations, it is crucial to make sure the models are trained on a variety of datasets.
- User Consent: Users must provide their informed consent after learning how and why their data will be used in order for the model to be trained.
- Misuse of Technology: Like any AI-powered technology, there is a chance that it could be abused. For example, the system might be used for surveillance or to monitor people using sign language without their consent.

6.4 Sustainability Plan

The following tactics ought to be used in order to guarantee the project's long-term viability and success:

- Constant Improvement: Technology should advance in tandem with sign language. For the system to remain relevant and helpful over time, it must be updated frequently to include new gestures, signs, or dialects that appear.
- Energy Efficiency: The project should concentrate on optimizing energy usage due to the computational needs of deep learning models. To operate the system effectively without leaving a significant environmental impact, this involves utilizing low-power hardware, cloud-based solutions, or more energy-efficient algorithms.
- Scalability: To ensure that the project can accommodate an expanding number of users and sign languages, it must be designed with scalability in mind. This could entail adding support for other sign languages to the system or creating portable, lightweight models.
- Cooperation with Stakeholders: In order to make sure that the project satisfies practical demands, it is essential to interact with advocacy organizations and the

deaf community. Working together with these organizations guarantees that technology will always be applicable, moral, and useful in resolving their issues.

6.5 Summary

In conclusion, by encouraging accessibility and inclusion, the Sign Language Recognition using Deep Learning project has the potential to greatly enhance the quality of life for those who use sign language. It has a huge social influence and helps create a more cohesive and sympathetic community. Even though there are certain ethical and environmental issues, a well-thought-out sustainability strategy that includes energy-efficient models, frequent updates, and stakeholder participation can guarantee the project's long-term success. By encouraging deeper understanding and communication across language barriers, this technology has the potential to create long-lasting beneficial change for the deaf and hard-of-hearing community as well as for society at large.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

By employing cutting-edge deep learning models, this project on Sign Language Recognition using Deep Learning Algorithms sought to close the communication gap for people who use sign language. We successfully created a system that can precisely interpret sign language motions by training models like DenseNet121, MobileNet, InceptionV3, VGG16, VGG19.

With an accuracy of 1.00%, the MobileNet model in particular beat the others, demonstrating the value of fusing the efficiency of InceptionV3Net with the dense connections of DenseNet. The research demonstrated great promise in fostering social inclusion and improving accessibility for the community of people with hearing impairments. The findings showed that deep learning may be successfully used to real-time applications such as sign language translation with proper tuning and optimization, significantly advancing the usage of AI in assistive technologies.

The integration of Streamlit-based API further ensured an interactive and a real-time control of this recognition model while ensuring the system was easy to use or adopt by ordinary users.

7.2 Further Suggested Works

Future improvements for enhancing both usability and effectiveness of the Sign Language Recognition System must address various aspects.

Expansion to Dynamic Gesture Recognition: The system today deals exclusively with static hand gestures. Recurrent Neural Networks (RNNs) and Transformers should be used to perform continuous sign recognition of entire sentences in future developments.

Multi-Language Sign Language Support: The model operates based on the information from American Sign Language (ASL). Implementing the system to recognize British Sign Language (BSL) and Indian Sign Language (ISL) and Chinese Sign Language (CSL) will enhance its accessibility.

Integration with Real-Time Speech-to-Text Conversion: Upgraded hardware will utilize TTS technology to transform detected gestures into instant speech so deaf and non-deaf communication can flow realistically.

Enhancement of Dataset with Real-World Variations: Models will achieve better generalization through the addition of extensive training data which includes people from different ethnicities and background hand shapes under diverse weather conditions.

Implementation on Edge Devices & Mobile Optimization: Using the model on lightweight electronic devices including smartphones, Raspberry Pi, or NVIDIA Jetson Nano allows wider accessibility across real-world situations. Mobile optimization through TensorFlow Lite or ONNX technology helps decrease processing demands without affecting performance quality.

Improved Noise Handling & Occlusion Reduction: The model can achieve higher robustness when it detects background noise and complex environments and partially obscures hands through depth camera integration or multi-modal sensor fusion which includes LiDAR and infrared.

Integration with Sign Language Translation Applications: The system functions as an integrated component in video conferencing tools and chat applications (such as Zoom, Google Meet and WhatsApp) to deliver real-time sign language interpretation benefits deaf populations.

7.3 Limitations

The project's advancement in sign language recognition through deep learning methods encounters multiple barriers which future development should resolve.

Limited to Static Gestures: The current operational model recognizes static hand gestures for individual writing letters and written words only. The system lacks the capability to comprehend both moving gestures and continuous strings of letters which are necessary components for interpreting entire sentences through sign language.

Language Restriction: The training data used for the model encompasses only American Sign Language (ASL). The system operates exclusively with American Sign Language (ASL) because it cannot recognize British Sign Language (BSL) or Indian Sign Language (ISL) which use distinct hand movements and linguistic patterns.

Environmental Sensitivity: The system's success rate might suffer due to insufficient lighting and background sounds as well as hand coverage during interpretation. Complex backgrounds together with poor illumination will lower the model's ability to recognize signs effectively.

Hardware and Computational Requirements: Both DenseNet and InceptionV3 deep learning models require powerful GPUs or cloud-based computation for their training functions and real-time detection processes which limit their deployment on basic hardware devices.

User-Specific Variability: The insufficient variety of samples in the dataset causes wrong classifications when dealing with size and color variations coupled with personal gestural styles.

Real-Time Processing Challenges: MobileNet delivers optimal performance for real-time applications yet complex models fail to achieve real-time processing capabilities especially when used on mobile devices because of the substantial computational requirements.

Lack of Facial Expression & Context Recognition: Sign language requires both facial expressions together with body movements for full meaning transmission. The present method processes hand gestures exclusively but this alone does not produce accurate interpretations for full sign language compositions.

Dataset Bias and Limited Training Data: The collection of data fails to present an accurate foundation which demonstrates all hand gestures that exist within natural environments. Generalization will increase through adding more varied hand postures alongside different skin colors and hand qualities to the training dataset.

References:

[1] A. Kumar, R. Gupta, and S. Sharma, "Sign language detection and recognition using deep learning," *Electronics*, vol. 11, no. 11, p. 1780, May 2022, doi: 10.3390/electronics11111780.

[2] D. Patel and M. Singh, "Sign language detection using deep learning," *Deepsign: Sign Language Detection and Recognition Using Deep Learning*, 2023. [Online]. Available: https://www.kaggle.com/datasets

[3] D. Patel, M. Singh, and K. Joshi, "Sign language recognition using modified deep learning network," *Deepsign: Sign Language Detection and Recognition Using Deep Learning*, 2023.

[4] J. R. Smith, L. K. Brown, and T. M. Chen, "Deep learning technology to recognize American sign language," *Sci. Rep.*, vol. 14, no. 1, p. 76174, Oct. 2024, doi: 10.1038/s41598-024-76174-7.

[5] S.Ali and H. Nguyen, "A comprehensive study on deep learning-based methods for sign language recognition," 2020.[Online].Available: https://arxiv.org/abs/2007.12530.

[6] M. Li, Z. Zhou, and X. Zhang, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," 2019. [Online]. Available: https://arxiv.org/abs/1910.11006.

[7] R. K. Yadav and P. S. Kim, "Mediapipe and CNNs for real-time ASL gesture recognition," 2023. [Online]. Available: https://arxiv.org/abs/2305.05296.

[8] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," 2020. [Online]. Available: https://arxiv.org/abs/2003.13830.

Appendix A

Course Outcomes, Complex Engineering Problems (EP) and Complex Engineering Activities (EA) Addressing

Title:
SIGN LANGUAGE RECOGNITION USING DEEP LEARNING

Student ID: [241-25-029]

CO Description for FYDP

CO	CO Descriptions	PO
Phase -I		
CO1	Integrate recently gained and previously acquired knowledge to recognize sign language for the Final Year Design Project (FYDP)	**PO1**
CO2	Analyze different aspects of the goals in designing a solution for this FYDP	**PO2**
CO3	Explore diverse problem domains through a literature review, delineate the issues, and establish these goals for the FYDP	**PO4**
CO4	Perform economic evaluation and cost estimation and employ suitable project management procedures throughout the development life cycle of the FYDP	**PO11**
Phase -II		
CO5	Design and develop technical solutions and system components or processes that meet specified requirements, ensuring compliance with public health and safety standards, as well as considering cultural, socioeconomic, and environmental factors in this FYDP	**PO3**
CO6	Choose and apply appropriate methodologies, resources, and contemporary engineering and IT technologies to address complex engineering processes, encompassing prediction and modeling, while adhering to relevant constraints in this FYDP	**PO5**

CO7	Analyze societal, health, safety, legal, and cultural considerations, along with associated responsibilities, in the context of professional engineering practice and the resolution of this problem, employing logical reasoning guided by contextual understanding.	**PO6**
CO8	Comprehend and evaluate the enduring sustainability and impact of professional engineering endeavors in addressing intricate engineering challenges within social and environmental frameworks.	**PO7**
CO9	Implement ethical principles and adhere to professional standards and norms in this FYDP	**PO8**
CO10	Capable of operating proficiently both individually and as a team member or leader across diverse teams and interdisciplinary settings in this FYDP.	**PO9**
CO11	Proficiently communicate with the engineering community and broader society regarding complex engineering endeavors, including the ability to comprehend and generate comprehensive reports and design documentation, as well as provide and receive clear instructions throughout this FYDP.	**PO10**
CO12	Acknowledge the importance of self-directed and life-long learning within the evolving landscape of technology, and possess the readiness and capability to engage in lifelong learning endeavors.	**PO12**

Addressing CO (1 to 8), Knowledge Profile (K), Attainment of Complex Engineering Problems (EP), and Attainment of Complex Engineering Activities (EA)

Addressing CO (1 to 8), Knowledge Profile (K), Attainment of Complex Engineering Problems (EP):

SN	EP Definition	Attainment	CO	Justification (with Knowledge Profile)	References
1.	**EP1: Depth of Knowledge required**	Yes	CO1, CO2, CO3, CO5, CO6, CO7 and CO8	Our project involves obtaining images hand for use in a dataset and building a deep learning model that includes the K3 and K4 aspects of the plant.	**Page no:** [3] **Section:** **[1.6]**
				The following details our solution: multiple models, system components, and web application covering K5 and K6.6.	**Page no:** [15-20] **Section:** [4.2, 4.3]
				We have considered with other models which exist at this moment with the same goal as we have (Work on sign language recognition while based on the image of the hand) which includes K8.	**Page no:** [4-7] **Section:** [2.2, 2.3]
2.	**EP2: Range of Conflicting Requirements**	Yes	CO2, and CO7	The appeared problem was to develop the dataset of hand image while we cannot get the satisfied regulations. This diversity signifies that the desired model should be designed to correctly identify the sign and should do this in a way that allows it to remain responsive to these differences. Thus, it addresses EP-2.	**Page no:** [9] **Section:** [3.2]

3.	**EP3: Depth of analysis required**	Yes	CO2, and CO6	I am going to explain that there are other ways that can be developed for recognize sign language. To decide which model would best examine how hand features differ with talking expression and the surroundings, we performed a detailed analysis, for our work.th talking expression and the environment and used it in our work. This addresses EP-3.	**Page no:** [15-19] **Section:** [4.3]
4.	**EP4: Familiarity of Issues**	Yes	CO8	When working on our sign language recognition project, we had to learn human biology and differentiate how a dumb makes himself and switch his signs from hand with the patrol and surrounding. At times we imposed upon ourselves to visit totally different areas such as ENT and Head-Neck Cancer Hospital & Institute-Dhaka which is not a topic included in our syllabus.	**Page no:** [1-2] **Section:** [1.2]
5.	**EP5: Extends of application codes**	No	CO5	N/A	N/A

6.	**EP6: Extends of stakeholders involved and conflicting requirements**	Yes	CO8	Our project includes numerous activities that are interrelated. These components include acquiring hand images, pre-processing data, training of deep learning models like DenseNet, MobileNet, InceptionV3, VGG16, VGG19 and develop a hybrid model, perform analysis to determine language and design a user interface web application.	**Page no:** [15-20] **Section:** [4.2, 4.3]
7.	**EP7: Interdependence**	Yes	CO5	There are interrelated subsystems in our paper. These responsibilities include gathering and processing data as well as training machine learning models and analyzing predictions.	**Page no:** [17-19] **Section:** [4.2, 4.3]
8.	-	Yes		We present a budget to evaluate and determine the costs required for and the revenues likely to be generated from the Final Year Design Project (FYDP)	**Page no:** **[13-14]** **Section:** **[3.4.2]**

Addressing CO11 with Complex Engineering Activities (EA) [Some or all the following]:

SN	EA Definition	Attainment	CO	Justification	References
1.	**EA1: Range of resources**	Yes	CO11	Our project uses multiple resources which include but not limited to high-performance computing, GPUs, deep learning frameworks, annotated datasets and ethical concerns to perform systematic work and enhance the classification of 26 signs from images using deep learning techniques especially through transfer learning and deep CNNs.	**Page no:** [9-11] **Section:** [3.2]
2.	**EA2: Level of interaction**	No		N/A	N/A

3.	**EA3: Innovation**	No		Our project propounds the brand-new web application featuring the new deep learning model combining the elements of both desenet and mobilenet that results in higher accuracy of hand image classification. Thus, the proposed application combines the advantage of both models and provides users with very effective and accurate tools to identify a language. Through state-of-the-art computing facilities, GPUs, and a rich annotated database, this web application provides a simple interface for anyone to use. Not only it enhances the field of ornithology, but it sets up new paradigms in implementation of artificial intelligence for conservation programs.	**Page no:** [41] **Section:** [5.3.3]

4.	**EA4: Consequences for society and the environment**	Yes		By providing improved sign language recognition for communicational research, using efficient rather than optimal computational power, and following the rules of data protection this project benefits society. Our initiative concerns environment: sustainability.	**Page no:** [43-45] **Section:** [6.1,6.2,6.3,6.4]
5.	**EA-5: Familiarity**	Yes		This work also proposes a new pipeline for sign language recognition based on the DeseNet and MobileNet models. This work specifically employs transfer learning and deep convolutional neural networks (CNNs) to present a comparative study as well as developing new pathways for developing ornithological research.	**Page no:** [21-41] **Section:** [5.2, 5.3]

Addressing CO (4, 9, 10, and 12):

SN	COs	Attainment	Justification	References
1	CO4	Yes	This proposed project contributes to the achievement of CO4 by incorporating appropriate use of project management and accurate financial control to enhance careful planning and control, estimation of resources for proper utilization as the lifeycle of the research project progresses.	**Page no:** [12-14] **Section:** [3.4]

2	CO9	Yes	The work is done ethically regarding plant leaf data privacy, receiving the required consent for data collection, and describing the research process. This guarantees the proper use of knowledge and societal welfare by moving in compliance with ethical principles and guidelines of illustrated small language recognition technologies cooperating with CO9.	**Page no:** [44] **Section:** [6.3]
3	CO10	No	N/A	N/A
4	CO12	Yes	As for the learning and adapting component (CO12) of this project within the dynamic field of the recognition of sign language, the collection of the data is meticulous, the statistical analysis and the methodological development of the approaches, and the highly detailed results and analyses derived from experiments are provided. This dedication demonstrates a prospective approach to keeping abreast of the information and furthering this understanding to enhance techniques for surviving present issues in ornithology.	**Page no:** [46-48] **Section:** [7.2,7.3]

241-25-029

ORIGINALITY REPORT

20%	14%	6%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	6%
2	github.com Internet Source	4%
3	Submitted to Higher Education Commission Pakistan Student Paper	1%
4	Submitted to CSU Northridge Student Paper	1%
5	pyimagesearch.com Internet Source	1%
6	H L Gururaj, Francesco Flammini, V Ravi Kumar, N S Prema. "Recent Trends in Healthcare Innovation", CRC Press, 2025 Publication	1%
7	Tyler, Jeramey. "Spatial Location of Binaural Signals Using Cepstral Analysis", Rensselaer Polytechnic Institute, 2024 Publication	1%
8	internship.daffodilvarsity.edu.bd Internet Source	<1%
9	Submitted to British University in Egypt Student Paper	<1%
10	Ulrich Agris. "Recent developments in visual sign language recognition", Universal Access	<1%