TASK 5

AMERICAN SIGN LANGUAGE (ASL) ALPHABET RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS

ARNAV GUPTA – RA2211028010107

SPANDAN BASU CHAUDHURI – RA2211028010120

AIM

The aim of this project is to develop a deep learning-based image classification model capable of recognizing hand gestures representing the **American Sign Language (ASL) alphabet** using Convolutional Neural Networks.

OBJECTIVES

1. To load and preprocess the ASL alphabet image dataset.
2. To resize, normalize, and label the dataset for model training.
3. To perform train-test splitting and one-hot encoding for classification.
4. To build a CNN model suitable for hand gesture recognition.
5. To train the model with early stopping to prevent overfitting.
6. To evaluate model performance using accuracy, classification report, and confusion matrix.

INTRODUCTION

Communication plays a vital role in human interaction, and sign language serves as an essential medium for individuals with hearing or speech impairments. The **American Sign Language (ASL)** alphabet consists of 26 hand signs with additional signs for *space*, *nothing*, and *delete*.

This project uses deep learning to automate ASL recognition. The model is trained on an image dataset where each image represents a hand sign corresponding to a particular alphabet letter. By extracting spatial patterns and gesture features, **Convolutional Neural Networks (CNNs)** learn to classify each sign accurately.

The dataset includes **29 classes**, representing:

- Letters **A–Z**
- Additional signs: **del**, **nothing**, **space**

REQUIREMENTS

Hardware Requirements

- 8GB+ RAM recommended
- GPU preferred (due to large dataset size and training time)

Software Requirements

Software/Library	**Version**	**Purpose**
Python	3.7+	Programming
TensorFlow / Keras	2.x	Model creation & training
OpenCV	Latest	Image loading & processing
Scikit-Image	Latest	Image resizing and transformations
NumPy & Pandas	Latest	Data manipulation
Matplotlib & Seaborn	Latest	Visualization

Dataset

- **ASL Alphabet Dataset** containing 87,000 color images labeled per sign.
- Each image resized to **64×64×3**.

LIBRARIES USED

Library	Purpose
NumPy, Pandas	Data handling
OpenCV & Skimage	Image loading & resizing
TensorFlow & Keras	Model training and evaluation
Matplotlib & Seaborn	Visualization and performance plots
Scikit-learn	Train-test split and evaluation metrics

ALGORITHM / METHODOLOGY

1. **Dataset Importing**
   - All images are loaded from labeled folders (A, B, C … space).
   - Each folder is mapped to a unique numerical label (0–28).
2. **Image Processing**
   - Images resized to **64×64**.
   - Pixel values normalized to range [0,1].
3. **Data Splitting**
   - Training and testing split: **70% train**, **30% test**.
   - Labels converted to **One-Hot-Encoding** for multi-class classification.
4. **Model Construction (CNN)**
   - Three convolutional layers + MaxPooling layers.
   - Flattening → Dense layer (128 neurons).
   - Output layer with **Softmax** activation for 29 classes.
5. **Model Compilation**
   - Optimizer: **Adam**
   - Loss: **Categorical Crossentropy**
   - Metric: **Accuracy**

6. **Early Stopping**
    - Stops training when validation loss stops improving.
7. **Model Training**
    - Epochs = 50 (early stopping may reduce this).
    - Batch size = 64.
8. **Evaluation**
    - Accuracy and loss curves plotted.
    - Classification report and confusion matrix generated.

INPUT / OUTPUT

Input	**Output**
Image of a hand sign	Predicted ASL character label

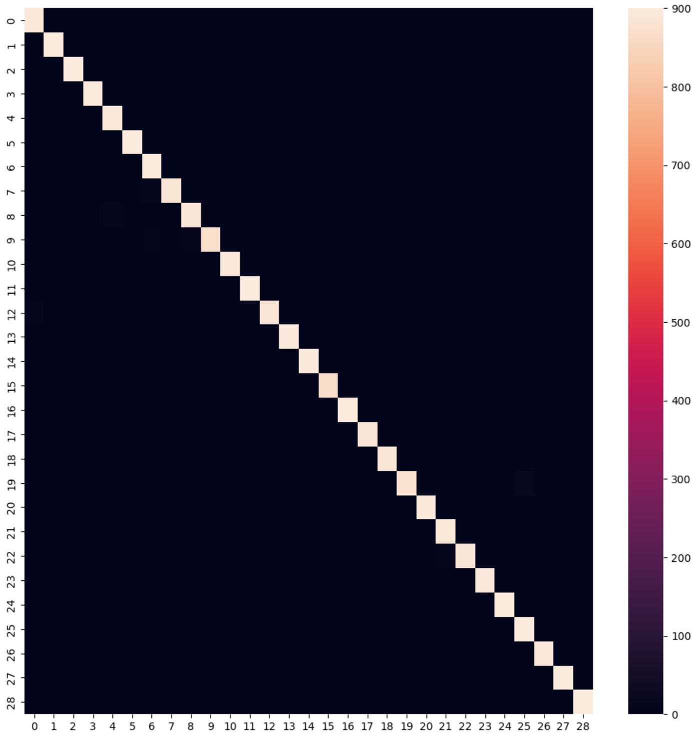
Example:

Input Image → Hand sign for “G”

Output → G

RESULTS

- The CNN successfully identified ASL gestures with high accuracy.
- Accuracy plots show stable improvement and minimized overfitting due to early stopping.
- Classification report showed strong performance across most classes.
- Confusion matrix indicated that visually similar signs may require additional feature enhancement, yet overall recognition was strong.



Class	Precision	Recall	F1-Score	Support
0	0.98	0.99	0.99	900
1	1.00	1.00	1.00	900
2	1.00	1.00	1.00	900
3	0.99	1.00	1.00	900
4	0.98	0.99	0.99	900
5	1.00	1.00	1.00	900
6	0.97	1.00	0.99	900
7	1.00	0.98	0.99	900
8	0.99	0.99	0.99	900
9	1.00	0.97	0.98	900
10	0.99	1.00	0.99	900
11	1.00	1.00	1.00	900
12	1.00	0.99	0.99	900
13	1.00	0.99	0.99	900
14	1.00	0.99	1.00	900

15	1.00	0.97	0.98	900
16	0.99	1.00	1.00	900
17	1.00	0.99	0.99	900
18	0.98	0.99	0.99	900
19	1.00	0.97	0.99	900
20	0.99	0.99	0.99	900
21	0.98	0.99	0.99	900
22	1.00	0.99	0.99	900
...	...	...	...	...

Metric	**Precision**	**Recall**	**F1-Score**	**Support**
Accuracy	—	—	**0.99**	26,100
Macro Avg	0.99	0.99	0.99	26,100
Weighted Avg	0.99	0.99	0.99	26,100

CONCLUSION

This project effectively demonstrates the use of **Convolutional Neural Networks for ASL Alphabet Recognition**. The model is capable of classifying 29 different hand signs with high accuracy after proper preprocessing and training techniques.

The system has potential for real-world use in:

- Sign-language translation applications
- Communication assistive technologies
- Educational learning tools

Future Enhancements:

- Convert model to real-time webcam-based ASL recognition.
- Apply transfer learning for robustness.
- Deploy the model to mobile platforms using TensorFlow Lite.