Slide 2: Introduction (1 minute)

My project focuses on developing a system to recognize American Sign Language (ASL) using Convolutional Neural Networks and deep learning.

The goal is to create a model that accurately identifies ASL gestures, which can help bridge communication gaps between ASL users and non-users, promoting inclusivity and accessibility.

I chose this project because of my passion for AI and machine learning and the significant social impact it can have. It presents a technical challenge that allows me to deepen my knowledge while contributing to a meaningful cause.

Additionally, this technology has future potential for advancements in assistive tools like real-time translation devices and better educational resources for the deaf community.

With that, let’s move on to the specifics of the methodology and techniques used in this project."

Slide 3: Introduction to ASL and Motivation (1 minute)

American Sign Language, or ASL, is a visual language used primarily by the deaf and hard-of-hearing community. It relies on hand gestures, facial expressions, and body language to communicate.

My motivation for this project is twofold:

First, the technological challenge. Working with CNNs and deep learning provides a great opportunity to apply advanced AI techniques to a real-world problem.

Accessibility: ASL is the primary language for many Deaf individuals, but it can be challenging for those who do not know sign language to communicate effectively with them. The development of ASL recognition systems can facilitate the bridging of this communication gap by enabling real-time translation of ASL gestures into text or spoken language.

➢ Inclusivity: ASL recognition systems can be valuable tools in educational settings, aiding Deaf students in accessing instructional materials and participating in classroom discussions. They can also be used by hearing individuals who are learning ASL to practice and improve their signing skills.

➢ Efficiency: Traditional methods of interpreting ASL, such as using human interpreters, can be time-consuming and costly. Automated ASL recognition systems offer a more efficient
Now, let’s dive into the specifics of the methodology and techniques used in this project."

Slide 4: Environment (1 minute)

Global Max Pooling2D:

It further condenses the feature map into a single vector, capturing the most prominent features.
Batch Normalization:

Applied to stabilize and accelerate training, ensuring consistent and efficient learning.
Dense Layer (256 units):

A fully connected layer with 256 neurons to learn complex patterns and representations from the extracted features.
Dropout:

This layer helps prevent overfitting by randomly dropping some neurons during training.
Dense Layer (29 units):

The final layer with 29 neurons, each corresponding to a class in the ASL dataset, using a softmax activation function to output probabilities for each class.
Output: The final classification result, indicating the recognized ASL gesture.

This structured approach ensures the model is both effective and efficient in recognizing ASL gestures.

Slide 6: Training and Testing the Model (1 minute)
I used the Adam optimizer for training because it combines the benefits of two other popular optimization methods, AdaGrad and RMSProp.
it is particularly effective for image data. Adam handles the high-dimensional and complex nature of images well by adjusting learning rates dynamically, leading to faster and more stable convergence.
Looking at the left graph, the training (red) and validation (green) losses decrease significantly over 50 epochs. This indicates that our model is learning effectively and generalizing well to unseen data.

Slide 7: Results and Interpretation (2 minutes)
Key Results: Precision, F1-Score, Confusion Matrix

Precision:

Our model achieved a precision score of approximately 0.79. Precision measures the ratio of true positive predictions to the total predicted positives, indicating that 79% of our model's positive predictions were correct.
F1-Score:

This user interface is crucial because it bridges the gap between complex machine learning algorithms and everyday users who may not have technical knowledge. It ensures that the model can be used by anyone, making it accessible and practical.

To summarize, we used Flask to create a web application that loads our trained ASL recognition model and provides an interface for users to upload images and get predictions. This deployment process makes our model usable in real-world scenarios, allowing users to recognize ASL gestures quickly and easily.

Thank you for your attention!

Slide 9: Future Directions and Conclusion (1 minute)

To further enhance the effectiveness and user experience of our ASL recognition system, we propose several future steps

1. Mobile Deployment:
   - Create mobile applications for iOS and Android to increase accessibility.
   - Optimize the model for mobile devices to ensure efficient performance and low resource consumption.
2. Continuous Learning:
   - Implement mechanisms for continuous learning where the model can be updated with new data over time.
   - Develop pipelines to collect new ASL gesture data from users to keep the model up-to-date.
3. Expand Vocabulary:
   - Extend the model to recognize more signs, including common words and phrases in addition to individual letters and basic commands.
   - Collaborate with ASL experts to ensure the inclusion of culturally and contextually relevant signs.
4. Community Engagement:
   - Engage with the ASL community to gather feedback and understand the practical challenges and requirements.
   - Use community insights to guide further development and refinement of the model and application.

In conclusion, our project on ASL recognition using CNN and deep learning shows great promise in helping the deaf and hard-of-hearing community communicate more easily.