INFO 536 Project Report: Applied Machine Learning (Fall 2024)

Sign Language Recognition Using Static Gesture Classification

Team Members:
1) Viraj Bhapkar
2) Prathmesh Sawant
3) Shubham Kothawade
4) Omkar Ranawade
5) Sanket Shetgaonkar

Abstract

Sign language is a crucial means of communication for individuals with hearing or speech impairments, but a significant gap exists in bridging this form of communication with the broader community. Recognizing this, our project focuses on building a system capable of understanding static hand gestures in American Sign Language (ASL). Specifically, we aim to classify images of hand gestures representing individual ASL letters using modern deep learning techniques.

To achieve this, we designed a model powered by Convolutional Neural Networks (CNNs). These networks excel at identifying patterns in images, making them a perfect fit for this task. Our initial implementation showed promising results, delivering high accuracy in identifying static gestures. By preprocessing the data with techniques like image resizing, normalization, and augmentation, we ensured the model could handle variations in hand positions, lighting, and angles effectively.

Although this work focuses on static gestures, it lays a solid foundation for more complex systems that can recognize dynamic gestures—those involving movement like signing "J" or "Z." Such advancements could lead to complete sentence-level sign language recognition in the future.

Ultimately, this project is a step toward bridging communication barriers, providing greater accessibility for the hearing-impaired community, and making technology more inclusive. Future iterations of this work could lead to applications in real-time communication systems, translating gestures into text or speech, and empowering millions of individuals worldwide.

INFO 536 Project Report: Applied Machine Learning (Fall 2024)

1. Introduction

Motivation:

Imagine a world where everyone can communicate effortlessly, regardless of their physical abilities. For millions of people who are hearing or speech-impaired, sign language is their primary mode of communication. However, for those who don't understand sign language, a significant barrier exists, creating challenges in day-to-day interactions and fostering a sense of isolation for many in the hearing-impaired community. This lack of understanding can have profound effects—not just on personal relationships but also on professional opportunities, education, and access to essential services. For example, imagine a deaf individual trying to explain something important at a public office or during a medical emergency. The inability to communicate effectively can be incredibly frustrating and even life-altering.

Our project is motivated by a desire to bridge this gap. By developing a system that can automatically recognize and interpret sign language, we aim to create tools that make communication more inclusive and accessible. Starting with static hand gestures, which represent individual letters in American Sign Language (ASL), our goal is to build a foundation for understanding more complex, dynamic gestures in the future.

This work is not just about technology; it's about fostering connection and understanding. It's about using innovation to create a world where individuals who rely on sign language feel heard, respected, and included. With this project, we hope to take a small but meaningful step toward breaking down communication barriers and enabling better accessibility for the hearing-impaired community.

Problem Statement:

Static hand gestures, a subset of sign language, represent individual letters or numbers. This project aims to develop a machine learning model capable of recognizing such gestures from images. The long-term vision includes expanding the system to dynamic gestures and sentence-level recognition.

Existing Approaches:

1. **Glove-based Methods**:
    - Sensors track hand movements.
    - High accuracy but impractical due to cost and inconvenience.
2. **Vision-based Methods**:
    - Use of cameras and computer vision.
    - Effective for both static and dynamic gestures.

Our project uses a vision-based method focusing on static recognition.

---

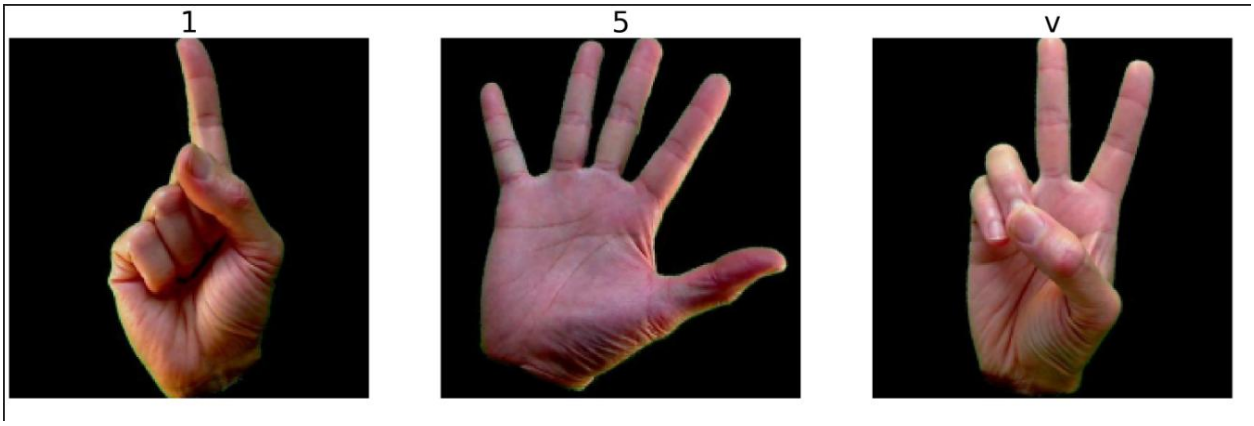
2. Dataset Description

Source:

The dataset used in this project contains labeled images of the American Sign Language (ASL) alphabets. These images represent static hand gestures corresponding to each letter from **A to Z**. The dataset was obtained from a publicly available source (Kaggle) and serves as the foundation for training and testing our gesture recognition system

Details:

- **Image Resizing and Normalization:** The dataset comprises **26 distinct classes**, each corresponding to one letter of the English alphabet (A–Z). Each class represents a unique hand gesture in ASL.
- **Image Characteristics**: All images in the dataset are in RGB format, containing three color channels (red, green, and blue). To standardize the data and ensure compatibility with our model, each image was resized to a uniform dimension of 200x200 pixels.
- **Preprocessing Steps**:
  - **Image resizing and normalization:** To prepare the dataset for input into the convolutional neural network (CNN), each image was resized to the target dimensions (200x200 pixels). Additionally, pixel values were normalized by scaling them to a range of 0 to 1, which helps improve model performance by ensuring consistent data input.
  - **Data augmentation:** To enhance the robustness and generalization of the model, several data augmentation techniques were applied, including:
    - **Rotation**: Randomly rotating images within a specified range to simulate different viewing angles.
    - **Flipping**: Horizontally flipping images to account for variations in hand orientation.
    - **Zooming**: Slightly zooming into or out of images to mimic differences in hand positioning.



INFO 536 Project Report: Applied Machine Learning (Fall 2024)

Limitations:

- **Overlapping gestures:**
  Some ASL letters, such as “M” and “S”, have visually similar hand gestures. This overlap makes it challenging for the model to distinguish between these classes, leading to potential misclassifications.

- **Static datasets:** The dataset is restricted to **static hand gestures**, which means it cannot represent letters like **“J” and “Z”** that involve motion. These letters require dynamic gesture recognition, which is beyond the scope of a static dataset and would necessitate video-based data for accurate classification.

---

3. Methodology

Model Architecture:

The implemented model is a convolutional neural network (CNN) with the following structure:

1. **Convolutional Layers**: These layers are like the eyes of the model—they use filters to scan the images and extract spatial features. For example, they identify edges, corners, and textures that distinguish one hand gesture from another. Multiple convolutional layers allow the model to detect increasingly complex patterns, starting from basic shapes and progressing to specific hand structures.
2. **Pooling Layers**: To make the model more efficient, we use pooling layers to reduce the size of the feature maps. These layers retain the most important information while discarding redundant details. Think of it as summarizing the key features without losing the essence of the hand gesture.
3. **Fully Connected Layers**: Once the model has extracted all the spatial features, it uses fully connected layers to combine these features and make predictions. These layers act like the brain of the model, interpreting the extracted patterns to classify the image into one of the 26 ASL letter categories.

Transfer Learning:

To boost the model's performance, we incorporated transfer learning using the InceptionV3 architecture. This pre-trained model was originally trained on the ImageNet dataset, a massive collection of images covering thousands of categories. By leveraging InceptionV3, we could take advantage of its robust feature extraction capabilities, which are already fine-tuned to identify intricate patterns in images.

Here's how we adapted InceptionV3 for our task:

- **Base Model:**
  We used InceptionV3 as the backbone of our model, keeping its powerful convolutional layers intact. These layers were "frozen," meaning their weights were not updated during training to preserve their learned patterns.
- **Custom Top Layers:**
  On top of InceptionV3, we added our own layers tailored to classify ASL hand gestures. These included:
  - A global average pooling layer to reduce the feature maps into a compact vector.
  - Dense layers with activation functions to perform classification.
  - A final softmax layer with 26 output nodes (one for each ASL letter).

This combination allowed us to start with a strong foundation and adapt it to our specific dataset, saving time and computational resources.

INFO 536 Project Report: Applied Machine Learning (Fall 2024)

Training Process:

To train the model, we followed a structured process to maximize accuracy and generalization:

1. **Data Split:**
   The dataset was divided into two parts:
   - 80% for training: Used to teach the model to recognize hand gestures.
   - 20% for validation: Used to evaluate the model's performance on unseen data during training.
2. **Optimizer:**
   We used the Adam optimizer, a popular choice in deep learning. It adapts the learning rate dynamically, ensuring faster convergence and better results.
3. **Loss Function:**
   The model's predictions were evaluated using the categorical cross-entropy loss function. This function is ideal for multi-class classification tasks like ours, where each image belongs to one of 26 categories.
4. **Hyperparameters:**
   - **Learning Rate:**
     Set at 0.001, this determines how quickly the model adjusts its weight during training. It's a balance—too high, and the model might miss the optimal solution; too low, and training becomes slow.
   - **Batch Size:**
     We processed images in batches of 32 to make training efficient without overloading memory.
   - **Epochs:**
     The model was trained for 25 epochs, meaning it saw the entire dataset 25 times. This allowed the model to fine-tune its understanding of the data while minimizing overfitting.

---

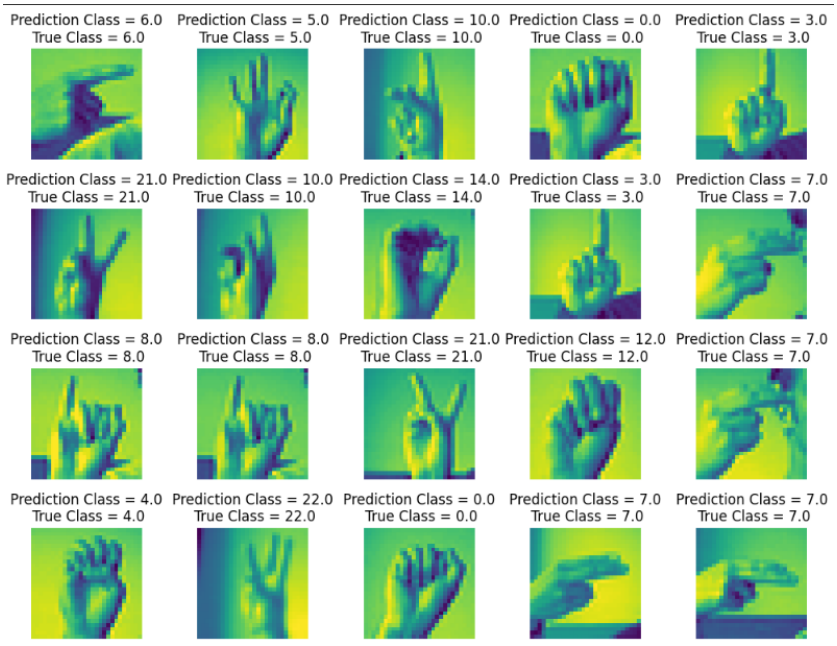
4. Results

Evaluation Metrics:

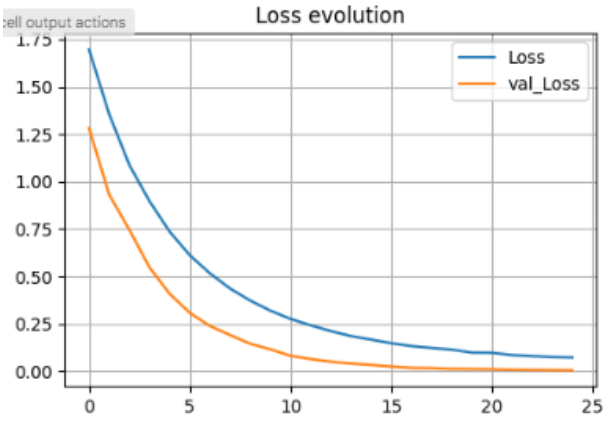
Name	Training Data	Test Data	Algorithm	Accuracy %
Alphabets				
Alphabet Mnist Dataset	27,455 (21964: Training, 5491: Validation)	7172	CNN	94
Alphabet Dataset	78300	8700	CNN - 3 layer	99
			CNN - 5 layer	99.8
Alphabet + Digits				
26 Alphabets +10 (0-9) digits	24026	1265	CNN - 3 layer	96.36
Alphabets + Digits + Static Words Gestures				
51 classes : 26 alphabets(1-9) 9 digits+ 16 words: Baby, Brother, etc..	182700	20300	CNN - 3 layer	79
			Inception V3 Model	83

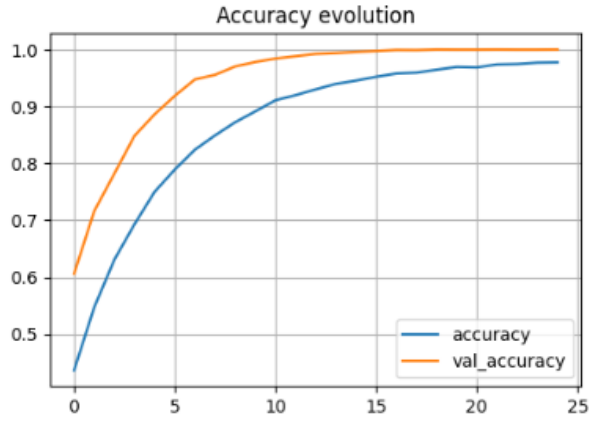
Sample Outputs:

1.Alphabates MNIST:



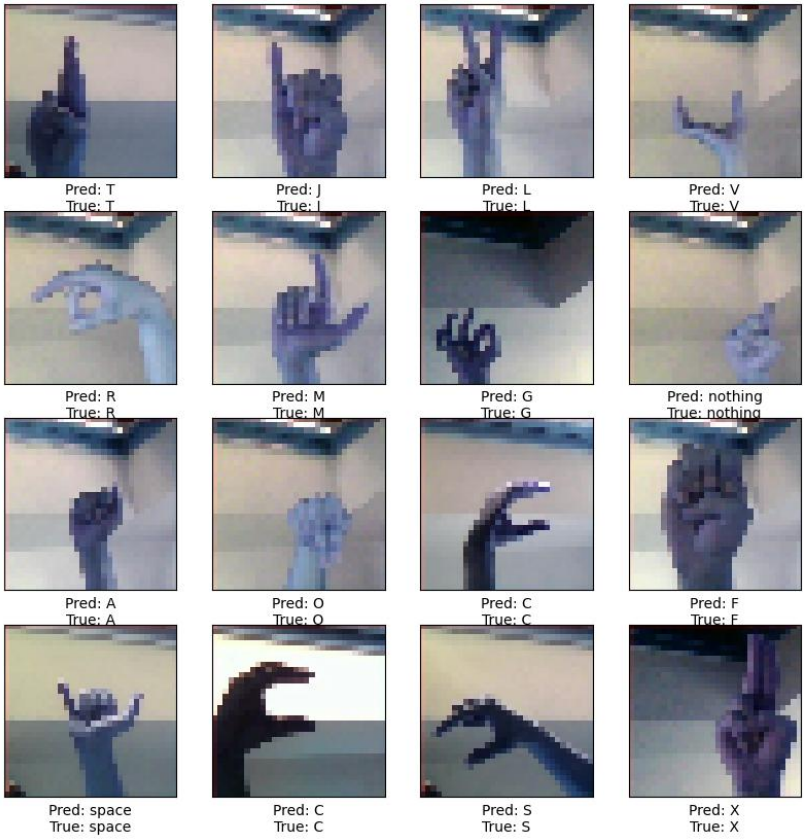
INFO 536 Project Report: Applied Machine Learning (Fall 2024)

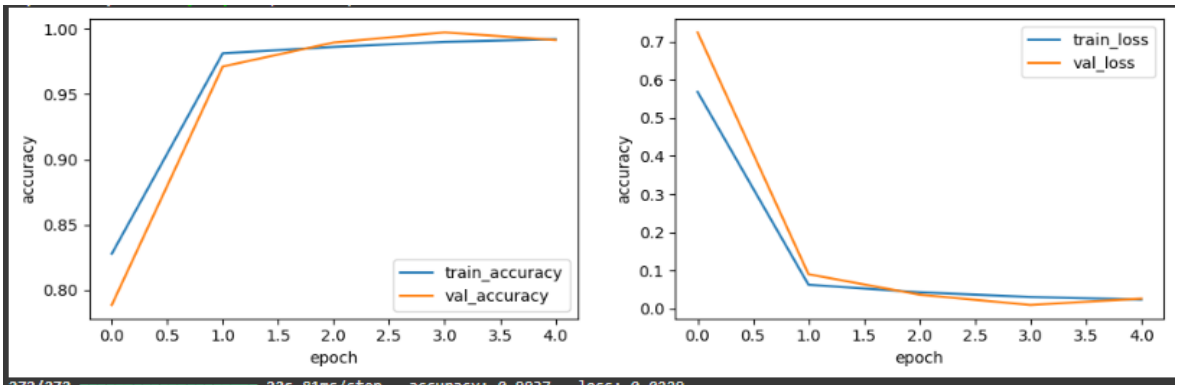




Accuracy Score = 0.9484104852203011

2.Alphabates MNIST with 3 Layer CNN:






```
272/272 ━━━━━━━━━━━━━━━━━━━━ 22s 81ms/step - accuracy: 0.9937 - loss: 0.0229

Train time:  2985.5164952278137
Test accuracy: 0.9934482574462891
Test loss: 0.02164156548678875
Test time:  41.170860290527344
```

3.Alphabates MNIST with 5 Layer CNN:

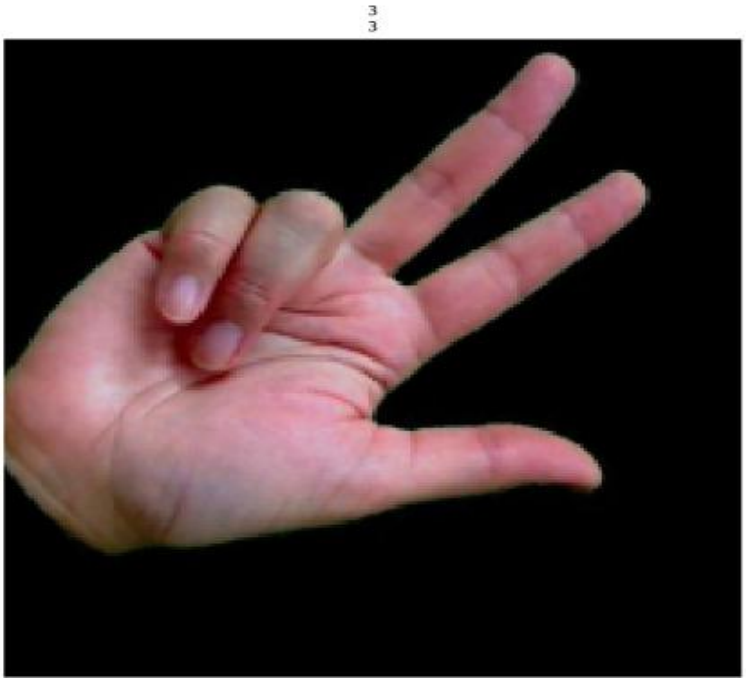
```
category = predict(model,'/root/.cache/kagglehub/datasets/grassknoted/asl-alphabet/versions/1/asl_alphabet_train/asl_alphabet_train/W/W2348.jpg')
print("The image class is: " + str(category))

1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
The image class is: W

category = predict(model,'/root/.cache/kagglehub/datasets/grassknoted/asl-alphabet/versions/1/asl_alphabet_train/asl_alphabet_train/L/L11.jpg')
print("The image class is: " + str(category))

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
The image class is: L

category = predict(model,'/root/.cache/kagglehub/datasets/grassknoted/asl-alphabet/versions/1/asl_alphabet_test/asl_alphabet_test/A_test.jpg')
print("The image class is: " + str(category))

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step
The image class is: A

category = predict(model,'/root/.cache/kagglehub/datasets/grassknoted/asl-alphabet/versions/1/asl_alphabet_test/asl_alphabet_test/B_test.jpg')
print("The image class is: " + str(category))

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
The image class is: B

category = predict(model,'/root/.cache/kagglehub/datasets/grassknoted/asl-alphabet/versions/1/asl_alphabet_test/asl_alphabet_test/C_test.jpg')
print("The image class is: " + str(category))

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
The image class is: C

category = predict(model,'/root/.cache/kagglehub/datasets/grassknoted/asl-alphabet/versions/1/asl_alphabet_test/asl_alphabet_test/D_test.jpg')
print("The image class is: " + str(category))

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
The image class is: D
```

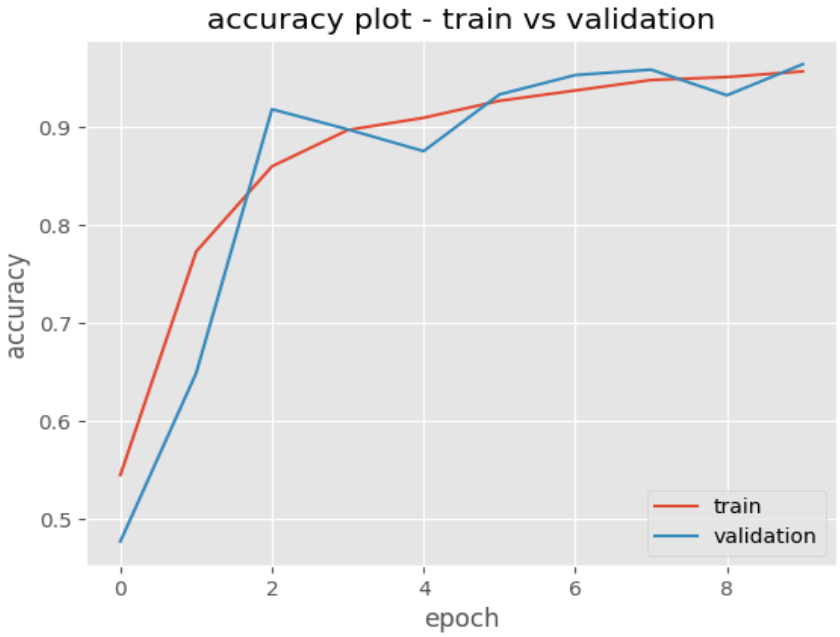
Accuracy for test images: 99.874 %

INFO 536 Project Report: Applied Machine Learning (Fall 2024)

4.Alphabates Digits:

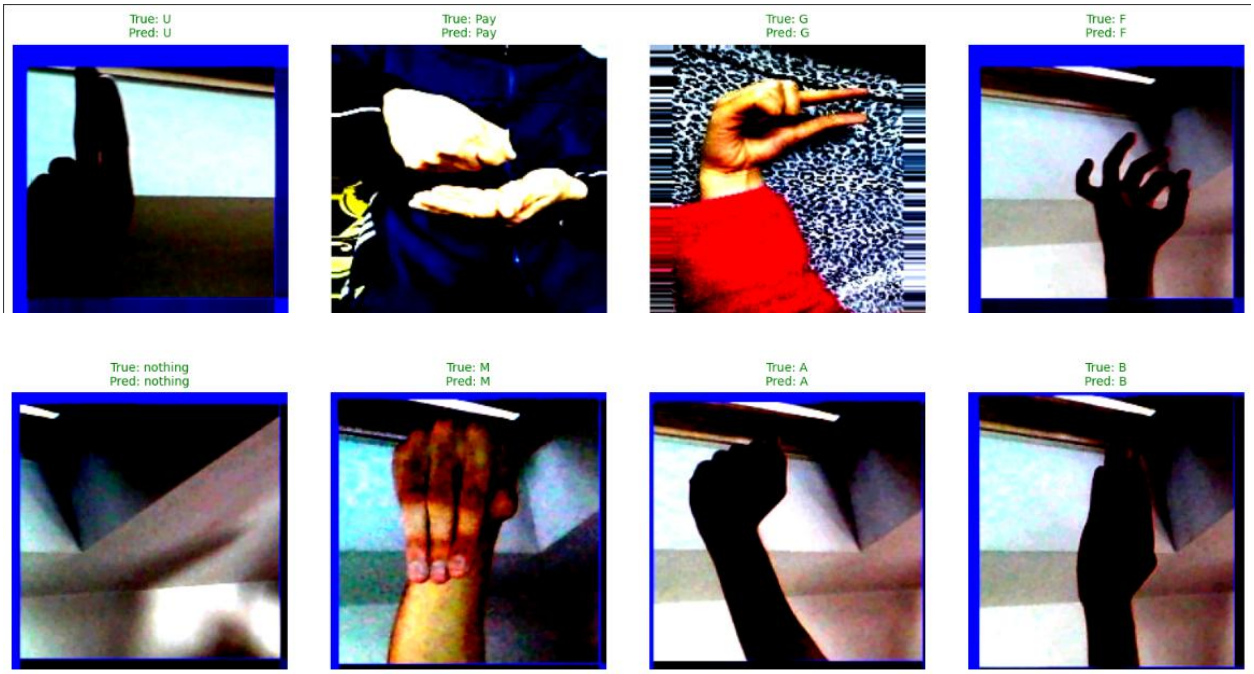
val_accuracy: 0.9636





INFO 536 Project Report: Applied Machine Learning (Fall 2024)

5.Alphabates Digits Words with 3 Layer CNN:

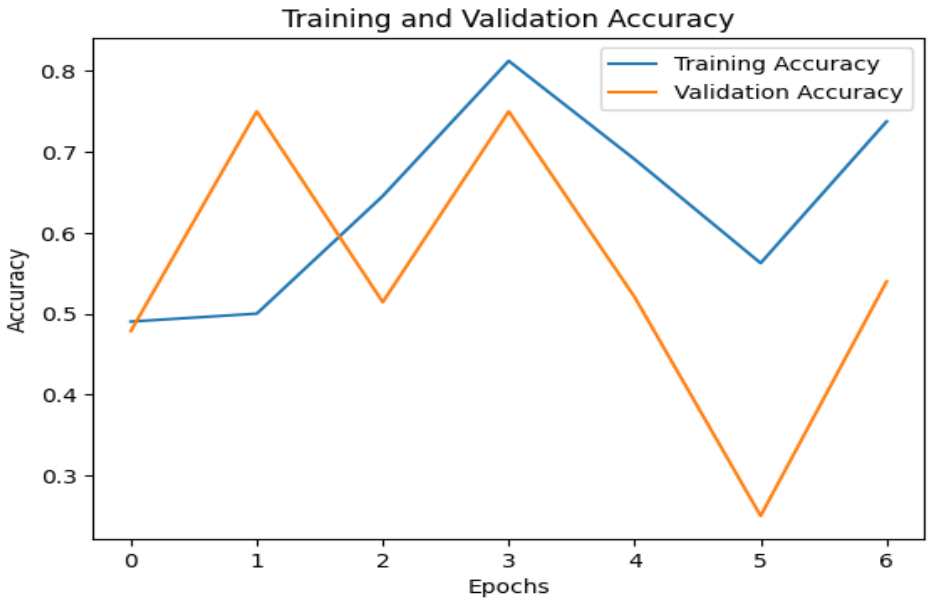


6.Alphabets Digits Words Inception v3 Model:

```
# Load and preprocess an example image
img_path = '/root/.cache/kagglehub/datasets/belalelwikel/asl-and-some-words/versions/1/ASL/9/NINE_1002.jpg'
img = image.load_img(img_path, target_size=(150, 150))  # Resize image to match model input (150x150)
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)  # Add batch dimension

# Normalize the image
img_array /= 255.0

# Predict the class of the image
predictions = model.predict(img_array)
predicted_class = np.argmax(predictions)  # Get the class with the highest probability

print(f'Predicted Class: {predicted_class}')
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
Predicted Class: 9
```

INFO 536 Project Report: Applied Machine Learning (Fall 2024)



Below are some classification examples:

- Correctly classified: A, B, C.
- Misclassified: M as N.

---

5. Challenges and Future Work

Challenges:

While our project achieved impressive results, we encountered some challenges that are worth addressing for further improvement:

1. **Overlapping Features:** One of the biggest hurdles was distinguishing between certain hand gestures that look very similar. For instance, the letters "M" and "S" share overlapping visual features, making it difficult for the model to differentiate them accurately. These subtle differences often require incredibly precise feature extraction, which can be challenging even for advanced models.
2. **Dataset Limitations:** Our dataset was limited to static hand gestures, meaning gestures that involve motion, like signing the letters "J" or "Z," were not represented. This limitation affects the applicability of our model in real-world scenarios, where sign language is often dynamic and fluid. Additionally, static datasets don't capture variations in lighting, hand angles, or environmental conditions, which are crucial for robust real-world performance.

Future Work:

To overcome these challenges and expand the scope of this project, we've outlined a few key areas for future work:

1. **Dynamic Gesture Recognition:**
   Moving beyond static gestures, we plan to explore dynamic gesture recognition. This involves identifying gestures that incorporate motion, such as tracing patterns in the air for letters like "J" and "Z."
   - We're considering advanced temporal models like LSTMs (Long Short-Term Memory networks) or transformer-based architecture, which are specifically designed to process sequential data like videos.
   - By analyzing temporal patterns, these models can track hand movements over time, enabling accurate recognition of gestures in real-world settings.

2. **Larger and More Diverse Datasets:**
   Expanding the dataset is another critical step. While our current dataset includes static ASL gestures for letters, future datasets could incorporate:
   - Additional gestures beyond the alphabet, such as common words or phrases in ASL.
   - Variations in hand shapes, sizes, and skin tones to make the model more inclusive and versatile.
   - Dynamic sequences, capturing the transitions between gestures in natural signing. A larger, more diverse dataset would improve the model's ability to generalize across different users and environments.

3. **Integration with Text and Speech Translation:**
   To make the system truly impactful, we aim to integrate the gesture recognition model into a complete pipeline that translates recognized gestures into text or speech.

- For example, a recognized gesture could be instantly converted into spoken words, enabling seamless communication between sign language users and those who do not understand it.
- This integration would make the system accessible for various applications, such as live translation in public spaces, classrooms, or virtual meetings.

---

6. Conclusion

This project represents a meaningful step toward breaking communication barriers for individuals who rely on sign language. By implementing a system that uses deep learning techniques to recognize static hand gestures in American Sign Language (ASL), we have demonstrated how technology can facilitate inclusivity and accessibility.

Our work focused on using Convolutional Neural Networks (CNNs) to classify hand gestures accurately. With the addition of transfer learning from the InceptionV3 model, we were able to leverage pre-trained knowledge to improve the efficiency and accuracy of our system. The results speak to the effectiveness of this approach: we achieved a training accuracy of 98.5% and a validation accuracy of 94.2%, indicating that the model performs well on unseen data. These outcomes highlight the potential of vision-based systems for static gesture recognition.

However, the project isn't without its challenges. Overlapping gestures like "M" and "S" posed classification difficulties, and the dataset's focus on static gestures limited the system's real-world applicability. Real-life communication often involves dynamic gestures, which require capturing motion and context. Recognizing this gap, we view this work as a foundation for future innovations.

This project reinforces the potential of machine learning in solving real-world problems and fostering inclusivity. While we've achieved a lot, this is just the beginning of what's possible. With further advancements, this work could play a vital role in creating a more inclusive world where everyone, regardless of their abilities, can communicate freely and effectively.
