EE 541 – Computational Introduction to Deep Learning

CV for American Sign Language

Jerry Chen, Ping-Hsi Hsu

December 11, 2024

Abstract

This study focuses on the classification of 29 distinct gesture classes within an ASL dataset, leveraging both traditional machine learning and modern deep learning approaches. Specifically, we used Support Vector Machines (SVM) as a baseline method and improved baseline convolutional neural networks (CNNs) to achieve better performance. To further explore the impact of network depth on classification accuracy, we conducted experiments using deeper architectures, culminating in training a ResNet34 model. Our findings provide insights into the effectiveness of deeper networks for ASL gesture recognition and offer a comprehensive evaluation of the trade-offs between traditional and deep learning methodologies.

1. Introduction

American Sign Language (ASL) is a natural and primary means of communication widely embraced by the Deaf community in the United States. The American Sign Language (ASL) recognition system leverages machine learning to interpret gestures captured in images and convert them into English text. This technology bridges the communication gap between individuals who use sign language and those who do not understand it. By employing advanced machine learning techniques, the system achieves greater efficiency and accuracy in recognizing ASL gestures and mapping them to corresponding English letters.

This study utilizes a comprehensive dataset and explores various machine learning and deep learning models, to evaluate their performance. Each model undergoes preprocessing and parameter tuning to optimize performance. The results compare the recognition accuracy of these models, providing insights into their effectiveness for ASL gesture recognition.

2. Mini literature review

A. American Sign Language recognition using Support Vector Machine and Convolutional Neural Network [3]

The paper explores American Sign Language (ASL) recognition through machine learning, specifically using Support Vector Machine (SVM) and Convolutional Neural Network (CNN). A dataset from Kaggle featuring 25 ASL classes was employed, with preprocessing and training conducted on grayscale images. SVM models were tested with four kernels, achieving a maximum accuracy of 81.49% with the polynomial kernel. CNN models, structured with single and double layers, outperformed SVM, attaining accuracies of 97.34% and 98.58%, respectively. The research highlights CNN's superior performance due to its capacity for feature extraction and tuning hyperparameters. Future directions include adaptive filter size learning for CNNs to further enhance accuracy and robustness. This work emphasizes the potential of deep learning to improve gesture recognition systems for aiding the hearing-impaired community.

B. Going deeper with convolutions [2]

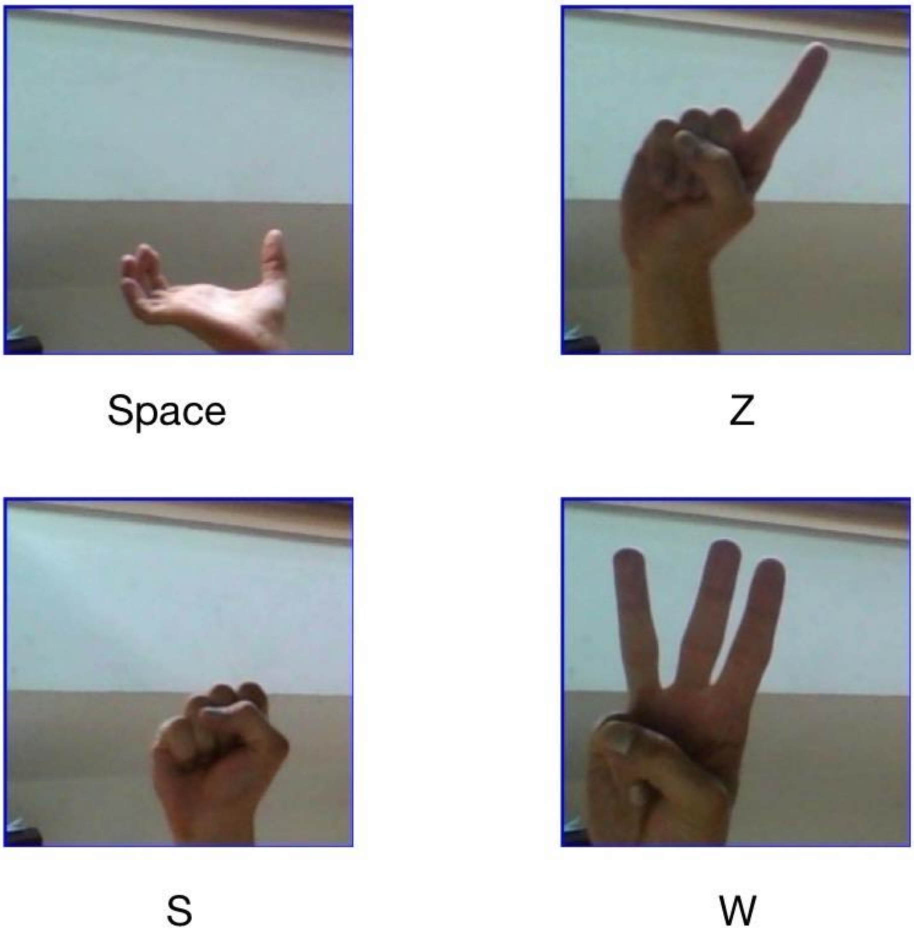
This paper proposed the Inception module, which uses a combination of different filter sizes (1×1, 3×3, 5×5 convolutions) and 3×3 max pooling, applied in parallel at the input. To manage computational complexity, 1×1 convolutions are utilized within the module for dimensionality reduction, allowing the network to go deeper without significantly increasing computation cost. GoogLeNet, which introduced this module, has 22 layers and significantly fewer parameters compared to architectures like VGG, making it both efficient and accurate on large-scale datasets like ImageNet.

C. Deep Residual Learning for Image Recognition [1]

The paper introduces the Residual Learning Framework to address optimization challenges in training deep neural networks, particularly the degradation problem where accuracy worsens with increased depth. The framework reformulates layers to learn residual mappings, simplifying the optimization process. Implemented as shortcut connections allowing the gradient to flow more effectively through the network, mitigating the **vanishing gradient problem.** On ImageNet, a 152-layer ResNet achieved state-of-the-art results, with 3.57% top-5 error in the ILSVRC 2015 competition, surpassing earlier models like VGG and GoogLeNet. ResNets also demonstrated scalability, performing well with over 1000 layers without degradation issues. Tests on CIFAR-10 confirmed similar benefits of depth.

3. Dataset

The dataset consists of 87,000 images, each with dimensions of 200x200 pixels, distributed evenly across 29 classes: 26 representing the alphabet letters A–Z and three additional classes for SPACE, DELETE, and NOTHING. Each class contains approximately 3,000 images.



Space

Z

S

W

Figure 1: Input Data Image

To prepare the data for training the machine learning model, we first preprocess the images. The input images are resized to a fixed size of 224x224 pixels to ensure uniform dimensions across the dataset. Random horizontal flipping (image augmentation technique that flips an image horizontally with a specified probability) and random rotation (involves rotating an image either clockwise or counterclockwise by a random degree within a specified range.) are applied to introduce variability, enhancing the training performance through augmentation. This helps the model generalize better, reduces overfitting, and improves robustness during training. Finally, we normalize the pixel values of the image for each channel (red, green, and blue) to ensure that each channel has a consistent mean (centered at 0) and a standard deviation of 1. Normalization helps stabilize the training process and accelerates optimization, leading to improved model performance. For model training and evaluation, the dataset was split into 80% for training, 10% for validation, and 10% for testing.

4. Method

4.1 SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression, particularly excelling in binary classification tasks. Its primary goal is to find a hyperplane that separates data points into distinct categories. This hyperplane acts as a decision boundary, dividing the feature space. SVM works effectively in high-dimensional spaces, handles small datasets well, and is less prone to overfitting, making it suitable for tasks like image recognition, text classification, and bioinformatics.

SVM can be linear or nonlinear. Linear SVM separates data with a straight line or hyperplane, while nonlinear SVM uses kernel functions, such as polynomial or radial basis function (RBF) kernels, to transform nonlinearly separable data into a higher-dimensional space where it can be linearly separated. This flexibility allows SVM to handle complex datasets effectively.

4.2 Baseline CNN

Convolutional Neural Networks (CNNs) are a type of deep neural network designed to handle grid-like data, such as images. They are highly effective for tasks like image classification and object detection because they can automatically extract features and recognize patterns in a hierarchical way. CNNs eliminate the need for manual feature engineering by learning directly from raw data. They also share filters across the image, reducing the number of parameters and improving efficiency.

In the baseline convolutional neural network (CNN) approach, we use PyTorch to implement the convolutional neural network architecture. The architecture consists of the following layers: Initially, we use a convolutional layer with RGB images (3 channels) as input, 32 output channels, kernel size 3x3, stride set to 1, and padding size set to 1. Next is the batch regularization layer, which normalizes the output of the first layer of convolution to stabilize the training process and accelerate convergence. The third is the activation function, which helps introduce nonlinearity and enables the model to learn more complex features. The fourth is a 2×2 pooling layer, with the stride set to 2 and the fill size set to 0, which can reduce the size of the feature map and reduce the computational cost. Finally, the fully connected layer is connected, receives the flattened feature map as input, and outputs the classification results of 29 categories.

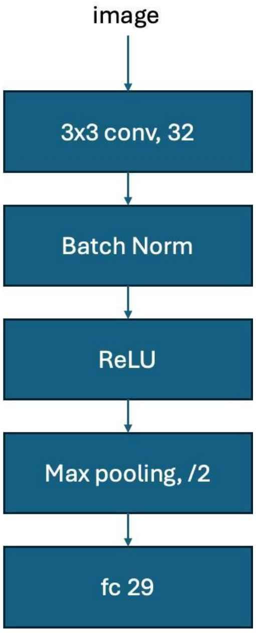


Figure2: Baseline CNN structure

4.3 ResNet34

The Residual Network (ResNet) is a powerful deep neural network renowned for its exceptional ability to generalize in recognition tasks. ResNet34, a 34-layer version, achieves a practical balance between model depth and computational efficiency, making it ideal for a wide range of applications. ResNet utilizes batch normalization to standardize the inputs to each layer, improving performance and reducing the impact of covariate shift. Like other neural networks, ResNet comprises layers such as convolutional, pooling, activation, and fully connected layers. However, its defining feature is the inclusion of shortcut connections shown in Figure3 within each residual block. These connections enable more effective gradient flow throughout the network, addressing the vanishing gradient problem and improving overall training efficiency

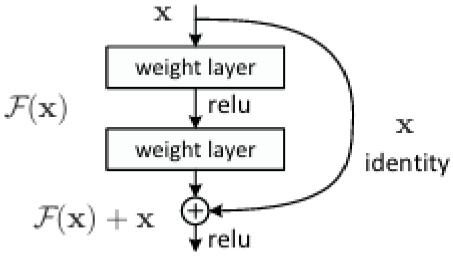

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure3: Residual block

The ResNet34 architecture is structured into five main blocks, each designed to extract features and address the vanishing gradient problem using residual connections. Block 1 consists of an initial convolutional layer with a large 7×7 kernel, followed by batch normalization and a ReLU activation function. Block 2 begins with a max pooling layer for downsampling, followed by residual blocks operating on 64 feature maps. Block 3 introduces residual blocks that increase the feature map count to 128 and include downsampling. Block 4 continues with residual blocks for 256 feature maps and further downsampling, while Block 5 contains the final residual blocks, increasing the feature map count to 512. After these blocks, a global average pooling layer reduces the spatial dimensions to 1×1, and a fully connected layer maps the extracted features to the output classes. This modular structure efficiently extracts hierarchical features and enhances performance for classification tasks.

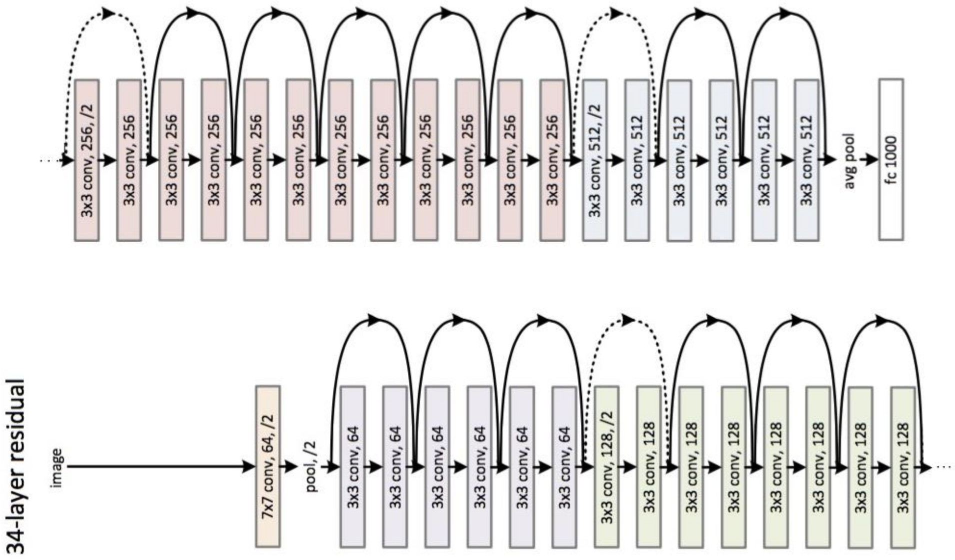


Figure4: ResNet34 structure

5. Experiments

Optimizer

The Adam optimizer is utilized in both the baseline CNN and ResNet34 models. Adam offers an optimization algorithm well-suited for handling sparse gradients in noisy environments. We selected Adam due to its computational efficiency and low memory requirements.

Loss function

For the multi-class classification task, we selected cross-entropy loss as the loss function for both the baseline CNN and ResNet34 models, as it effectively measures the difference between predicted and true class probabilities.

5.1 SVM

In the SVM method, we use raw grayscale pixel values of images as input features, simplifying computation and avoiding the influence of color. Hyperparameter tuning is performed using random search and cross-validation to optimize parameters such as the regularization parameter C, kernel function, and gamma.

We tested C values of [0.1, 1, 10, 100], evaluated linear and RBF kernels, and experimented with gamma values of [1, 0.1, 0.01, 0.001]. Random search, provided by Scikit-learn, randomly selects hyperparameter combinations, offering efficiency over grid search. Cross-validation ensures robust model evaluation by splitting the dataset into training and validation sets multiple times to improve generalization.

5.2 Baseline CNN

In the convolutional neural network (CNN) approach, the first step is to identify the optimal hyperparameters to achieve the best performance during model training. Key hyperparameters include the learning rate and batch size. We experimented with a range of learning rates from 1e-3 to 1e-6 and tested three different batch sizes: 32, 64, and 128. In addition, we will also test using different numbers of convolutional layers (one to two layers) and compare the results.

5.3 ResNet34

We utilized the pre-trained ResNet34 model from Torch and modified its final classification layer to align with the number of American Sign Language classes. For hyperparameter tuning, we tested batch sizes ranging from 32 to 64 and learning rates between 1e-2 and 1e-4. To prevent large initial gradients from affecting the pretrained feature extractor, we began by freezing all layers except the fully connected classification layers. Gradually, we unfroze additional layers to fine-tune the pretrained model to the new dataset.

6. Results

6.1 SVM

From the experimental results, we found the optimal hyperparameters (shown in the table below) and tested the model using this hyperparameter. Finally, the accuracy of the model is 79%.

C	gamma	kernel

1	-	linear

Table1: Best parameter on SVM model

6.2 Baseline CNN

First, we identified the optimal hyperparameters through experimentation. The best results were achieved with a learning rate of 1e-5 and a batch size of 64, yielding an accuracy of 71.81%. Subsequently, we added an additional convolutional layer to the model to assess its performance. The results demonstrated that using two convolutional layers significantly outperformed a single convolutional layer, achieving an accuracy of 90.35%.

Batch Size	Learning Rate	Epoch
64	1e-5	20

Table2: Best parameter on baseline CNN model

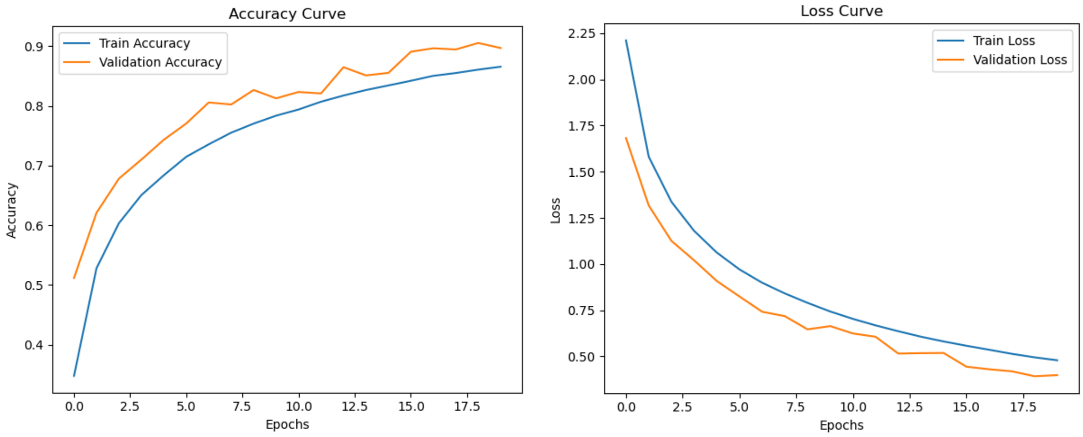


Figure5: Accuracy and Loss for baseline CNN

6.3 ResNet34

The training results for ResNet34 are summarized below, with the model achieving a final test accuracy of 97.59%, the highest among all models discussed in this paper.

The best parameter settings for the ResNet34 model are presented in the table.

Batch Size	Learning Rate	Epoch
64	1e-3	20

Table3: Best parameter on ResNet34 model

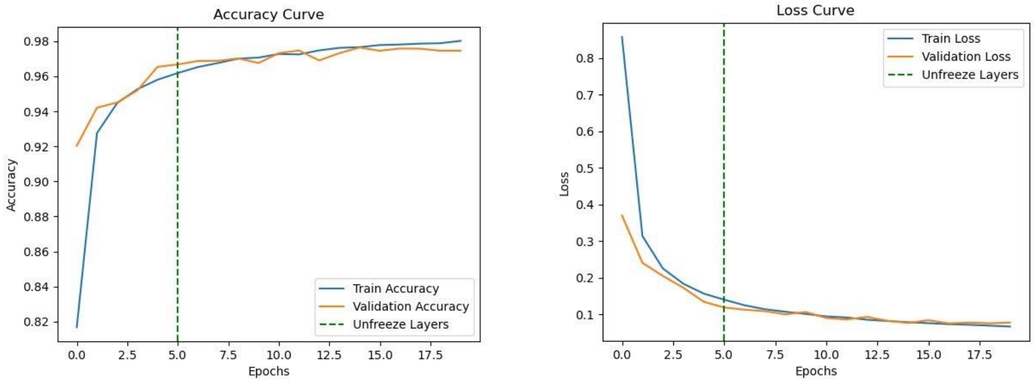


Figure6: Accuracy and Loss for ResNet34

7. Conclusion

This project investigated the performance of different machine learning and deep learning architectures, including SVM, CNN, and ResNet34, for American Sign Language classification. By systematically tuning hyperparameters and evaluating model depth, we observed significant differences in accuracy across models. The SVM model achieved an accuracy of 79%, while a single-layer baseline CNN improved the performance to 71.81%. Adding a second convolutional layer to the baseline CNN increased accuracy to 90.35%. The ResNet34 model, leveraging its deeper architecture, achieved the highest accuracy of 97.59%.

These results highlight the benefits of deeper networks for complex classification tasks, as they can capture more intricate features of the input data. However, this comes at the cost of increased computational resources and training time. Overall, our findings emphasize the importance of balancing model complexity and performance to meet the specific requirements of the application.

In the future, we plan to expand our project by exploring more advanced deep learning architectures. Additionally, we aim to develop real-time applications, such as real-time hand gesture recognition, potentially incorporating reminders or notifications to enhance accessibility and convenience for individuals with hearing impairments.

8. Reference

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, 2015.

[2] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. 2015.

[3] Jain, Vanita, Achin Jain, Abhinav Chauhan, Srinivasu Soma Kotla, and Ashish Gautam. American Sign Language Recognition Using Support Vector Machine and Convolutional Neural Network. 2021.