Deep Learning Final Project
Sign Language (ASL) to Text Translation

Wenying Li
2020080108
liwenyin20@mails.tsinghua.edu.cn

Boqiao Wang
2021012245
wangbq21@mails.tsinghua.edu.cn

Yingwei Shi
2023010694
syw23@mails.tsinghua.edu.cn

Abstract

Sign language is a vital communication tool for the hard-of-hearing and deaf communities, yet its interpretation by non-signers remains a challenge. This project focuses on developing a deep learning-based system to translate American Sign Language (ASL) gestures into text, fostering more inclusive communication. Leveraging an open-source ASL alphabet dataset with 87,000 images spanning 29 classes, we implement and compare two models: a Convolutional Neural Network (CNN) and StarNet.

The CNN model serves as a baseline, offering simplicity and robust performance, while StarNet introduces a lightweight yet powerful architecture for enhanced feature extraction. Both models are trained with data augmentation and optimized using the Adam optimizer with categorical cross-entropy loss. Evaluation metrics such as accuracy, precision, recall, F1-scores, and confusion matrices assess model performance and pinpoint areas for improvement.

Our results highlight the potential of deep learning in accurately recognizing and classifying ASL gestures, paving the way for real-world applications. This work contributes toward bridging communication gaps and promoting accessibility for the hard-of-hearing community.

Keywords: *American Sign Language (ASL), Gesture Recognition, Convolutional Neural Network (CNN), StarNet, Image Classification*

1. Introduction

Communication is a fundamental aspect of human interaction, yet for individuals in the hard-of-hearing and deaf communities, bridging the gap between sign language and spoken or written languages remains a significant challenge. American Sign Language (ASL), a visual language comprising hand gestures, facial expressions, and body movements, is widely used among the deaf community. However, its adoption and understanding among non-signers are limited, often leading to barriers in everyday communication. This project aims to address this challenge by developing a deep learning-based system that translates ASL gestures into text, enabling more inclusive and seamless communication between sign language users and non-signers.

The project focuses on leveraging advances in computer vision and natural language processing (NLP) to create an efficient and accurate ASL-to-text translation model. By combining image classification techniques with robust preprocessing methods, the system aspires to recognize ASL gestures and convert them into meaningful text outputs. Such a tool holds the potential to empower the hard-of-hearing community by making their interactions more accessible and inclusive in real-world scenarios.

2. Related Work

Recent advances in sign language recognition leverage Convolutional Neural Networks (CNNs) for automatic feature extraction, replacing traditional methods that relied on handcrafted features and classifiers like SVMs and KNNs. Kumar et al. (2022) [1] proposed a system combining CNNs with OpenCV for real-time ASL gesture recognition. The system captures hand gestures via webcam, applies preprocessing (grayscale conversion, background subtraction), and feeds them into a CNN trained on a custom dataset. Data augmentation, including resizing and normalization, improves model generalization. The model achieved 99% training accuracy, 100% validation accuracy, and 97% test accuracy, demonstrating strong generalization to unseen data. Unlike prior models that only worked with static images, this system supports real-time gesture recognition, providing an interactive tool for seamless communi-

cation between sign language users and non-signers.

In 2022, LI et al. [3] designed a novel Transformer-style module, the CoT module, for visual recognition. This design fully leverages the contextual information between input keys to guide the learning of dynamic attention matrices, thereby enhancing the capability of visual representation. The CoT module first encodes the input keys with contextual information through a 3×3 (k × k) convolution, obtaining a static contextual representation of the input. Then, the encoded keys are concatenated with the input queries and two consecutive 1×1 convolutions are used to learn the dynamic multi-head attention matrices. The learned attention matrices are multiplied with the input values to achieve a dynamic contextual representation of the input. Finally, the fusion of static and dynamic contextual representations is used as the output. CoTNet-50/101 and CoTNeXt-50/101 achieved better performance compared to existing visual backbone networks.

In 2024, Xu et al. [2] proposed starnet The star operation in neural network design has untapped potential, capable of mapping inputs to high-dimensional nonlinear feature spaces, similar to kernel tricks, without increasing network width. StarNet achieves efficient feature representation through its unique "star operation" (element-wise multiplication). This operation can map inputs to high-dimensional nonlinear feature spaces within a compact network structure and low energy consumption, without increasing computational complexity. While maintaining computational efficiency, StarNet can obtain richer and more expressive feature representations. Additionally, StarNet has the characteristic of low latency, which is particularly important for applications with high real-time requirements. The StarNet model achieved excellent performance on the ImageNet-1k dataset.

3. Approach

This project employs two deep learning models—Convolutional Neural Networks (CNN) and StarNet—for ASL gesture recognition and translation. These models were chosen for their demonstrated performance in image classification tasks and their ability to balance accuracy with computational efficiency. CNN serves as a baseline model due to its simplicity and effectiveness, while StarNet is explored for its advanced feature extraction and lightweight architecture. YOLO, while powerful for object detection, was not used as it is less suited for single-hand gesture classification tasks where bounding box localization is unnecessary.

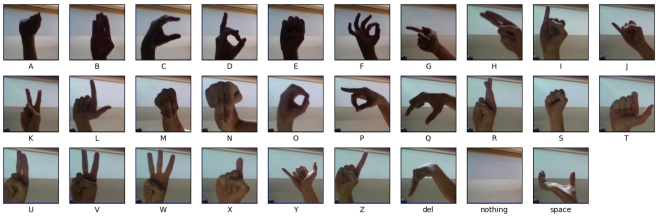

Figure 1. Kaggle ASL Dataset

4. Experiment

4.1. Dataset

We are using an open-source ASL alphabet dataset from Kaggle, which includes approximately 87,000 images of 29 classes. Of these, 26 classes represent the letters A-Z, while 3 additional classes for—SPACE, DELETE, and NOTHING—are essential for real-time applications and classification. The test data set contains a mere 29 images, to encourage the use of real-world test images. See figure 1.

4.2. Convolutional Neural Networks (CNN)

For our baseline model, we use convolutional networks, as they show reasonable performance in the related works. This CNN model architecture uses a series of convolutional pooling, and fully connected layers.

4.2.1 Model Architecture

The proposed CNN architecture consists of two 2D convolutional layers, each with 32 filters of size 3x3 and ReLU activation, followed by 2x2 max-pooling layers that reduce spatial dimensions while preserving essential features. After the convolutional blocks, a flatten layer transforms the 2D feature maps into a 1D vector, which feeds into a fully connected layer with 128 neurons and ReLU activation. To prevent overfitting, a dropout layer with a rate of 0.5 randomly deactivates half of the neurons during training. Finally, a softmax-activated output layer with 29 units predicts class probabilities for each of the 29 classes. See the model architecture diagram below:

4.2.2 Training and Optimization

The model is compiled using the **Adam optimizer** and the **categorical cross-entropy loss function** to handle multi-class classification. The model is only trained on training dataset which is 90% of the whole dataset, and training is performed for 15 epochs with a batch size of 32.

4.3. Starnet

In the single-layer of neural networks, the star operation is usually written as (W1X+B1) * (W2X+B2), which means

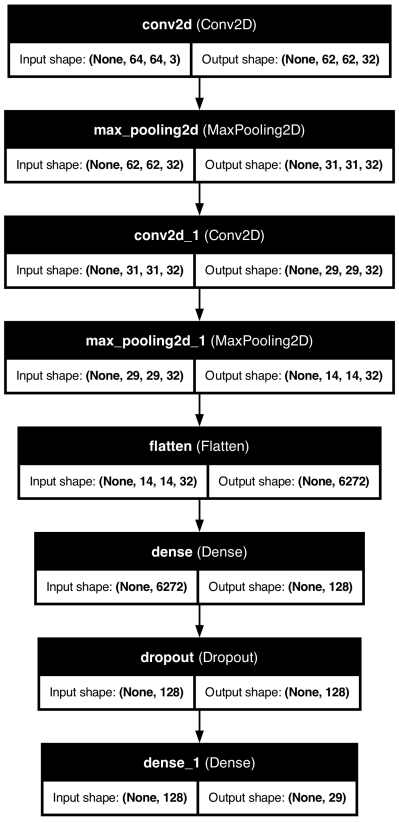

Figure 2. CNN Model Architecture

$$
\begin{aligned}
& w_1^{\mathrm{T}} x * w_2^{\mathrm{T}} x \\
= & \left(\sum_{i=1}^{d+1} w_1^i x^i\right) * \left(\sum_{j=1}^{d+1} w_2^j x^j\right) \\
= & \sum_{i=1}^{d+1} \sum_{j=1}^{d+1} w_1^i w_2^j x^i x^j \\
= & \underbrace{\alpha_1^1 x^1 + \cdots + \alpha_{(4,5)} x^4 x^5 + \cdots + \alpha_{(d+1,d+1)} x^{d+1} x^{d+1}}_{(1,1)}
\end{aligned}
$$

Figure 3. rewrite the stars

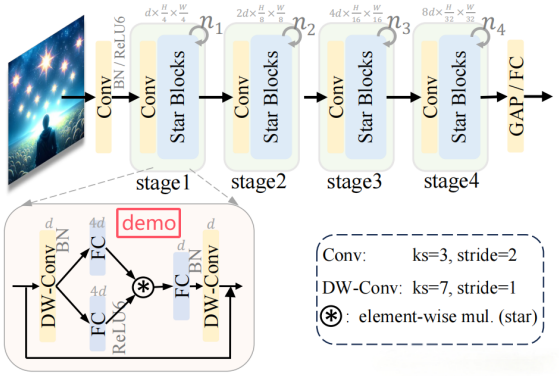

Figure 4. Starnet

fusing the features of two linear transformations through element wise multiplication.

Rewriting the original star operation can expand it into a combination of two different items, as shown in Figure 3 It is worth noting that, except for one item, each item exhibits non-linear correlation, indicating that they are independent and implicit dimensions. Therefore, we use star operations with high computational efficiency to perform calculations in d-dimensional space, but can achieve representation in the implicit dimensional feature space of d2/2, significantly enlarging the feature dimension, while requiring any additional computational overhead in a single layer. This significant characteristic shares a similar concept with kernel functions.

Expanding single-layer star operations to multiple layers can obtain the basic units of starnet. As shown in Figure 4, the author designed the original StarNet as a four-stage hierarchical structure, using convolutional layers for downsampling and modified demo blocks for feature extraction. Layer Normalization is replaced with Batch Normalization and placed after depthwise convolution (which can be fused during inference). Inspired by MobileNeXt, depthwise convolutions are added at the end of each block. The channel expansion factor is consistently set to 4, with the network width doubling at each stage. The GELU activation function in the demo block is replaced with ReLU6, following the design of MobileNetv2. Different sizes of StarNet are constructed by changing the number of blocks and the number of input embedding channels.

Our network adopts the structure of the original network. To make it lightweight, the depth is set to [1,2,6,2] with embed numbers of 32, and the output category is set to 29. The training process is similar to before.

4.4. Results and Analysis

4.4.1 CNN

By training the CNN model for 15 epochs, the accuracy and loss on the training set indicate excellent performance. The final training accuracy reaches 99.60%, and the training loss is 0.0116 as shown in Figure 5. Preliminary analysis shows no significant signs of overfitting, suggesting the model generalizes well to the training data.

Testing the model on a validation or test dataset is necessary to confirm its robustness. However, the current results indicate that the CNN architecture performs exceptionally well for ASL gesture recognition tasks, making it a strong candidate for real-world applications.

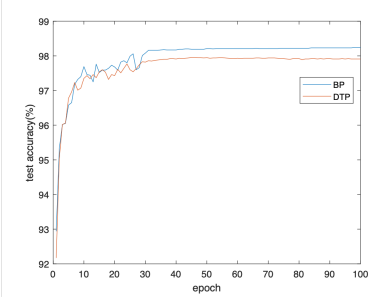

Figure 5. loss and accuracy of CNN

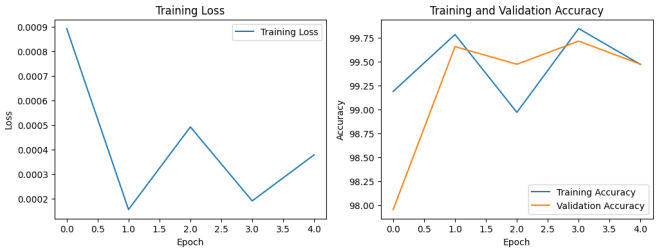

Figure 6. loss and accuracy of Starnet

4.4.2 Starnet

Setting the learning rate to 0.001 and training for 5 rounds, the accuracy of the loss function on the training and testing sets is shown in Figure 6. The final accuracy on the testing set reaches 99.47%, and there is basically no overfitting phenomenon, indicating that Starnet has good performance in this task.

4.4.3 Discussion

We compared two deep learning models, CNN and Starnet, for real-time conversion of American Sign Language (ASL) gestures into text. We did not use any pre trained models or weights, and trained from scratch on the ASL dataset. CNN showed a testing accuracy of 99.6%, while Staenrt's accuracy reached 99.4%. There was no significant difference in accuracy between the two. Considering the integration and deployment with Mediapipe on local devices in the future, Starnet may be more suitable for practical applications due to its lower latency and ability to train inference on CPUs.

Our research is still not comprehensive and does not include a series of classic networks such as ViT, Convnext, Mobilenet, etc. At the same time, we have not had the opportunity to integrate with Meidapipe and deploy it on local devices. Future work can study more models, further explore ways to improve model accuracy, and how to maintain minimal performance degradation in deploying real-time tasks.

5. Conclusion and Future Works

In this project, we explored the potential of deep learning models to accurately recognize and classify American Sign Language (ASL) gestures into text. While the CNN model offers simplicity and reliable performance, StarNet's lightweight architecture and computational efficiency make it particularly suitable for deployment on resource-constrained devices, such as mobile phones or edge computing systems. However, further testing on real-world data is required to evaluate the models' performance beyond the controlled dataset environment.
Looking ahead, we find three promising directions:

1. **Experimentation with Advanced Architectures.** Investigate state-of-the-art models like Vision Transformers (ViT), ConvNeXt, and MobileNet for their applicability to ASL gesture recognition.

2. **Real-time ASL-to-Text Converter Application.** Create an interactive application that visualizes the ASL-to-text translation process in real time.

3. **Diverse and Real-World Dataset Expansion.** Incorporate additional ASL datasets that include variations in lighting, backgrounds, and user hand shapes to improve the model's generalizability.

6. Acknowledgement

Members' relative contributions in % to the project:

- Wenying Li: 45%. CNN Model, Report.
- Boqiao Wang: 45%. Starnet Model, Report.
- Yingwei Shi: 10%. Poster.

References

[1] Hemant Kumar, Mohit Kumar Sharma, Rohit, Kunal Singh Bisht, Ashish Kumar, Rachna Jain, Preeti Nagrath, and Pritpal Singh. Sign language detection and conversion to text using cnn and opencv. 2022. AIP Conf. Proc. https://doi.org/10.1063/5.0108711. 1

[2] Xu Ma, Xiyang Dai, Yue Bai, Yizhou Wang, and Yun Fu. Rewrite the stars. 2024. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 5694-5703. 2

[3] Dai Qin, Gao Yingcai, Wang Hongjiang, and Shen Qingze. Defect recognition of photovoltaic panels based on adaptive dimensional feature aggregation convolutional neural network. 2024. Journal of Physics Conference Series 2806(1):012016 DOI:10.1088/1742-6596/2806/1/012016. 2