Arabic Sign Language Classification using Custom ResNet

Ahd Osama
ID: 2305296

Omar El-Sayed
ID: 2305057

Mohamed El-Sayed
ID: 2305273

Yasser Ali
ID: 2305268

Mohamed Ashraf
ID: 2305181

Moataz Ahmed
ID: 2305223

Abstract

Arabic Sign Language (ArSL) recognition is a critical task for bridging the communication gap for the deaf and hard-of-hearing community. In this paper, we propose a custom Deep Learning approach to classify the 28 letters of the Arabic alphabet from hand gestures without the use of transfer learning. Our architecture, ArSL-ResNet, utilizes a custom ResNet-18 backbone integrated with Squeeze-and-Excitation (SE) blocks to provide channel-wise attention, focusing on critical hand features. To improve robustness and resolve confusion between visually similar signs like ZAY and SEEN, we implemented MixUp augmentation and label smoothing. Our experimental results demonstrate that this combination significantly improves generalization, achieving a final test accuracy of 98.28%, surpassing our initial baseline of 95.86%.

1 Introduction

Arabic Sign Language (ArSL) serves as the primary medium of communication for the deaf and hard-of-hearing community across the Arab world. However, a significant communication gap exists between signers and non-signers. Automated recognition of ArSL hand gestures using Computer Vision is a vital step toward bridging this gap, enabling real-time translation and more inclusive digital environments.

Despite recent advances in Deep Learning, ArSL recognition presents unique challenges. The Arabic alphabet consists of 28 letters, many of which share highly similar hand postures. For instance, the letters **ZAY** and **RAA**, or **SEEN** and **SHEEN**, differ only by subtle finger movements or positions. Furthermore, variations in lighting, background clutter, and individual signer styles make it difficult for standard models to generalize well.

In this work, we aim to achieve >95% accuracy without using pre-trained models (Transfer Learning). Training from scratch requires an architecture capable of deep feature extraction without falling into the vanishing gradient trap. We propose a custom ResNet-18 integrated with Squeeze-and-Excitation (SE) blocks. Furthermore, we address the model's initial struggle with ambiguous classes by introducing MixUp augmentation, which forces the model to learn smoother decision boundaries, ultimately reaching a test accuracy of 98.28%.

2 Dataset and Data Preparation

The dataset used is the "Arabic Sign Language Unaugmented Dataset," comprising a total of 5,832 images, each with a size of 416×416 pixels, representing the 28 letters of the Arabic alphabet.

2.1 Data Structure and Challenges

The raw data was provided in YOLO format. To make this compatible with our ResNet architecture, we implemented a custom pre-processing script to:

1. Parse the YOLO label files to extract only the **Class ID**, discarding the spatial coordinates.
2. Organize images into a directory structure: `train/`, `val/`, and `test/`, with subfolders for each of the 28 classes.
3. Resize all images to a uniform 128×128 resolution.

Light Variation

Messy Background

2.2 Data Augmentation

To ensure the model did not overfit to specific backgrounds, we applied a strong augmentation pipeline during training, including random horizontal flips, rotations (up to 20°), color jittering (brightness and contrast), and affine translations.

3 Proposed Methodology

To achieve high-accuracy classification without transfer learning, we designed a custom architecture based on the Residual Network (ResNet) framework, enhanced with channel-wise attention mechanisms.

3.1 The Degradation Problem and Vanishing Gradients

In traditional deep convolutional neural networks (CNNs), increasing the depth (number of layers) theoretically allows the model to learn more complex features. However, as depth increases, accuracy often saturates and then degrades rapidly—a phenomenon known as the **degradation problem**. This is not caused by overfitting, but by the difficulty of optimizing deeper layers due to the **vanishing gradient problem**.

During backpropagation, the gradient is calculated via the chain rule, multiplying partial derivatives layer by layer from the output back to the input. Mathematically, if these derivatives are small (< 1), their product decays exponentially as they travel backward.

Consequently, the gradients for the initial layers become nearly zero, meaning their weights are never updated. In the context of ArSL, the model effectively "forgets" how to extract foundational features—such as the edge orientation of the hand—making it impossible for deeper layers to build complex gesture representations upon them.

3.2 How ResNet solves this problem

ResNet solves this by introducing Identity Shortcut Connections. Instead of forcing the network to learn a direct mapping $H(x)$, we recast the task as learning a residual mapping $F(x) := H(x) - x$. The output of a residual block is defined as:

$$y = F(x, \{W_i\}) + x$$

The addition of the identity term x has a profound mathematical implication: the gradient can now flow through two paths. Even if the weight layers $F(x)$ suffer from vanishing gradients, the shortcut path ensures that the gradient of the identity term (which is 1) allows information to pass unimpeded.

ArSL Intuition: The shortcut path preserves global hand geometry (silhouettes and edges), while the residual path focuses on learning the subtle finger variations that distinguish one letter from another. This dual-focus prevents the model from losing foundational structural features as network depth increases.

3.3 Squeeze-and-Excitation (SE) Blocks

While ResNet preserves spatial information, it lacks a mechanism to prioritize different feature channels. In sign language recognition, channels capturing finger-tip positions are high-value, while those capturing background textures are noise. We integrated **SE Blocks** to add "Channel Attention."

The SE block operates in two stages:

1. **Squeeze ($\boldsymbol{F}_{\text{sq}}$):** Global Average Pooling condenses 2D spatial information into a 1D channel descriptor $z \in \mathbb{R}^C$. This represents a global "summary" of what each channel has detected across the whole image:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j)$$

2. **Excitation ($\boldsymbol{F}_{\text{ex}}$):** A bottleneck mechanism learns the non-linear dependencies between channels to produce a set of weights s:

$$s = \sigma(W_2 \ \text{ReLU}(W_1 z))$$

ArSL Intuition: The final output is scaled by s, effectively "turning up the volume" on channels that contain discriminative features (like the gap between fingers) and "muting" channels that represent irrelevant information.

This dynamic prioritization is what allows our custom model to achieve high precision without pre-training 28 different gestures.

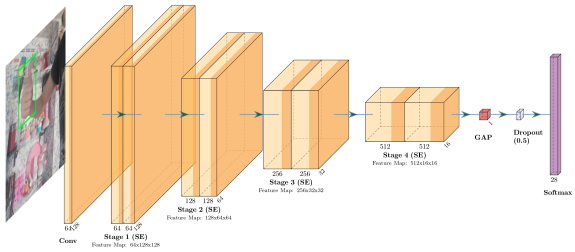
4 Model Architecture

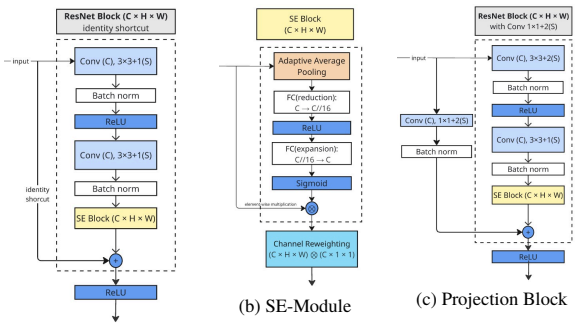
ResNet-18 was selected over deeper variants such as ResNet-50. While deeper models can capture more complex abstractions, a 50-layer network would introduce excessive parameter capacity (~23.5 million parameters), leading to **overfitting** on the ArSL dataset. By utilizing ResNet-18, we maintained an optimal balance between model depth and generalization.

4.1 Network Topology

The model processes $128 \times 128 \times 3$ input tensors through the following hierarchical stages:

- **Initial Feature Extraction:** A 3×3 convolutional layer with 64 filters and Batch Normalization.
- **Residual Backbone:** Four sequential stages, each containing two Residual Blocks. The filter depth doubles at each stage ($64 \rightarrow 128 \rightarrow 256 \rightarrow 512$).
- **SE-Integration:** Every residual block incorporates a Squeeze-and-Excitation module after the second convolution to recalibrate channel-wise feature responses.
- **Classification Head:** A Global Average Pooling (GAP) layer condenses the 512 feature maps, followed by a 0.5 Dropout layer and a Dense Softmax layer.





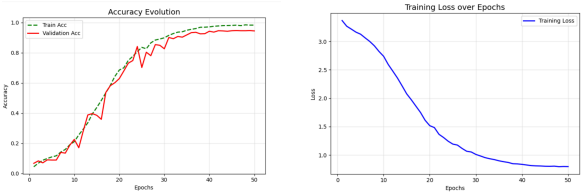
(a) Identity Block (b) SE-Module (c) Projection Block

4.2 Hyperparameters

Training was executed over 60 epochs to ensure full convergence.

We utilized the **Adam** optimizer rather than **AdamW** to mitigate the risk of **over-regularization** since MixUp already functions as a powerful structural regularizer.

The learning rate was regulated by a **Cosine Annealing** scheduler, initialized at $\eta = 1e-3$ and decaying to a minimum of $\eta = 1e-6$ to refine the weights in the final epochs.



4.3 Phase 1: Baseline Performance and Failure Analysis

Initially, the model was trained using standard geometric augmentations(rotation, flipping, and scaling). While it achieved an impressive **95.86%** accuracy, a confusion matrix analysis revealed "over-confidence" in visually similar gestures. Specifically, the model struggled with the **ZAY** and **SEEN** classes, which share high structural overlap in low-resolution feature maps.

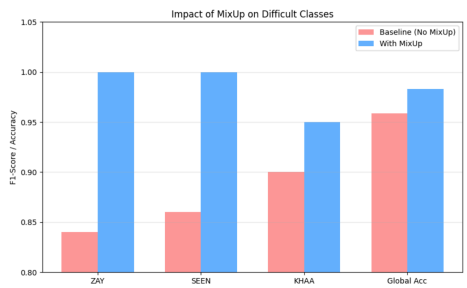
DHELL	1.00	1.00	1.00	10
RAA	0.91	1.00	0.95	10
ZAY	0.89	0.80	0.84	10
SEEN	0.82	0.90	0.86	10
SHEEN	1.00	1.00	1.00	11

4.4 Phase 2: Boundary Smoothing via MixUp

To address this, we transitioned to a MixUp-based training. By training on convex combinations of image pairs, we forced the model to learn a continuous feature space rather than discrete, memorized points.

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \quad \tilde{y} = \lambda y_i + (1-\lambda)y_j$$

This adjustment proved transformative. By softening the labels, the model developed more robust decision boundaries that effectively separated the previously confused classes.



4.5 Final Performance Summary

The final evaluation on the test set demonstrated the superiority of the MixUp-enhanced SE-ResNet.

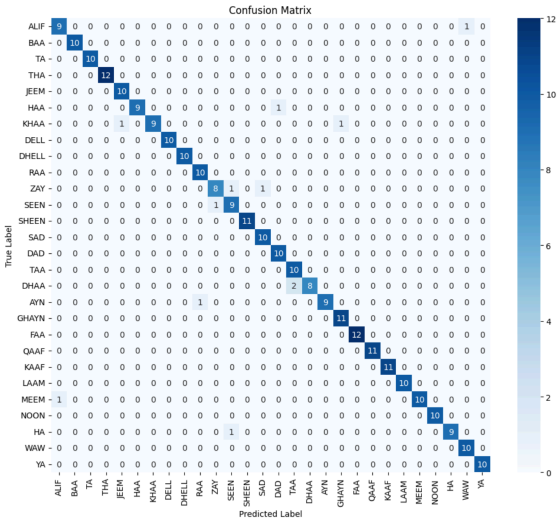
Class	Phase1 (Baseline) F1	Phase2 (MixUp) F1	Improvement
ZAY	0.84	1.00	+16%
SEEN	0.86	1.00	+14%
KHAA	0.90	0.95	+5%

```
Final Test Accuracy: 98.28%

Classification Report:
              precision    recall  f1-score   support

        ALIF       1.00      1.00      1.00        10
         BAA       1.00      1.00      1.00        10
          TA       1.00      1.00      1.00        10
         THA       1.00      1.00      1.00        12
        JEEM       0.91      1.00      0.95        10
         HAA       0.91      1.00      0.95        10
        KHAA       1.00      0.91      0.95        11
        DELL       1.00      1.00      1.00        10
       DHELL       0.91      1.00      0.95        10
         RAA       1.00      1.00      1.00        10
         ZAY       1.00      1.00      1.00        10
        SEEN       1.00      1.00      1.00        10
       SHEEN       1.00      1.00      1.00        11
         SAD       0.91      1.00      0.95        10
         DAD       1.00      1.00      1.00        10
         TAA       1.00      0.90      0.95        10
        DHAA       0.91      1.00      0.95        10
         AYN       1.00      1.00      1.00        10
       GHAYN       1.00      0.91      0.95        11
         FAA       1.00      1.00      1.00        12
        QAAF       1.00      1.00      1.00        11
        KAAF       1.00      1.00      1.00        11
        LAAM       1.00      1.00      1.00        10
        MEEM       1.00      0.91      0.95        11
        NOON       1.00      1.00      1.00        10
          HA       1.00      1.00      1.00        10
         WAW       1.00      1.00      1.00        10
          YA       1.00      0.90      0.95        10

    accuracy                           0.98       290
   macro avg       0.98      0.98      0.98       290
weighted avg       0.98      0.98      0.98       290
```



4.6 Example Predictions

5 Conclusion

Our paper confirms that a custom-designed SE-ResNet, when paired with modern regularization techniques like MixUp, can achieve state-of-the-art results (98.28%) on Arabic Sign Language without the need for Transfer Learning. The combination of channel attention and boundary smoothing provides a robust framework for handling the high visual complexity of the Arabic alphabet.