Enhanced Image Analysis Techniques Using Multiscale Feature Extraction and Attention Mechanisms

Achraf Kamni (A20509594)
Sakher Yaish (A20496906)

November 14, 2024

Abstract

This project explores a framework to improve image recognition methods by enhancing the ability to capture intricate details and contextual information. Building on the approaches from Balat et al. (2024) and Garg et al. (2024), we adapted and modified key techniques to create a more effective system. Our approach incorporates multiscale feature extraction, spatial-reduction attention, and progressive dimensional reduction to retain essential features while emphasizing important regions within static images. Through comparative analysis across several model architectures, we illustrate the strengths of our methods using visual and quantitative metrics, showing the potential of our improvements over baseline approaches.

1 Introduction

1.1 Problem Statement

In today's image analysis, many methods struggle to capture small details and important context, which affects their accuracy in recognizing images. Traditional techniques often aren't flexible enough to focus on the most important parts of a static image, so they miss chances to improve accuracy. Also, most current approaches don't handle different levels of image features very well, which limits their performance in various situations.

Recently, there's been more interest in creating better image recognition systems, especially for recognizing non-verbal communication like sign language. Arabic Sign Language recognition is particularly challenging because it requires careful understanding of complex gestures. Our project aims to build a framework that improves image recognition, especially for Arabic Sign Language. Using techniques like multiscale feature extraction, spatial-reduction attention, and progressive dimensional reduction, we aim to create a system that keeps the important parts of an image while focusing on key regions. This way, we can get better recognition results without losing important details.

1.2 Objective

Our main objectives are:

1. Implement multiscale feature extraction to capture both detailed and contextual features within images.
2. Integrate spatial-reduction attention mechanisms to enhance the model's focus on important regions of each image.
3. Apply progressive dimensional reduction to maintain a structured hierarchy of features, preserving essential information.
4. Compare our model's performance with established architectures, including ResNet, Vision Transformer (ViT), and GoogleNet, to evaluate the effectiveness of our approach.

2 Background

2.1 Transformers in Computer Vision

Transformers, initially introduced for natural language processing tasks [1], have been successfully adapted for computer vision applications [2]. The key component of transformers is the self-attention mechanism, which allows the model to weigh the importance of different parts of the input data.

2.2 Multiscale Attention Mechanism

The multiscale attention mechanism enables the model to capture features at various scales, addressing challenges such as variations in object size, pose, and shape. By progressively reducing the embedding dimensions across transformer stages, the model can focus on both fine-grained details and global context.

2.3 Related Work

Balat et al. (2024) [4] explored advanced Arabic alphabet sign language recognition using transfer learning and transformer models, demonstrating the potential of transformers in image recognition tasks. Garg et al. (2024) [3] proposed the Multiscale Video Transformer Network (MVTN) for hand gesture recognition in videos. Our work expands upon these concepts, adapting techniques from video recognition to static image analysis and incorporating multiscale feature extraction for improved model accuracy.

3 Methodology

3.1 Approach and Models

We implemented several models to analyze and improve the recognition of Arabic Sign Language from grayscale images. The process and findings for each model are described below:

3.1.1 Basic U-Net

The U-Net architecture was selected as our baseline model. Known for its encoder-decoder structure, U-Net is commonly applied to tasks requiring localization, such as segmentation. Here, we adapted U-Net to process grayscale images of Arabic Sign Language gestures, using its simple architecture as a starting point.

3.1.2 Fine-Tuned ResNet-50

Next, we employed a ResNet-50 model, which is known for its depth and residual connections that mitigate vanishing gradient issues in deep networks. The ResNet-50 model was fine-tuned on our dataset to capture more complex and nuanced features required for distinguishing between subtle sign language gestures.

3.1.3 Fine-Tuned GoogleNet

Following ResNet-50, we trained GoogleNet, a deeper model with multiple inception modules that allow the network to learn at different receptive fields. The model was fine-tuned on our dataset, with the expectation that its inception structure would allow it to capture more intricate gesture details.

3.1.4 Fine-Tuned Vision Transformer (ViT)

The Vision Transformer (ViT) model was also employed for this task. Unlike convolutional networks, ViT leverages self-attention mechanisms to process image data in patches, potentially allowing it to capture spatial relationships and complex patterns. The model was configured with batch visualizations to observe the learning patterns.

3.1.5 Custom Model with Pretrained ResNet-18

We developed a custom model incorporating a pretrained ResNet-18 as a feature extractor. By leveraging pretrained weights, we aimed to transfer learned representations from a larger dataset to facilitate recognition in our specialized task. This approach allowed us to assess the efficacy of pretrained features in recognizing sign language gestures.

3.1.6 Custom Model with Non-Pretrained ResNet-18

Lastly, a custom model based on a ResNet-18 architecture without pretrained weights was implemented. By training the model from scratch, we evaluated how the absence of pretrained weights affected the learning process, which provided insights into the role of pretrained feature extraction in our classification task.

3.2 Proposed Approach

We propose a series of advanced techniques for image analysis that build upon the ideas in the referenced papers while introducing novel elements. Our approach includes:

1. **Multiscale Feature Extraction**: Implement a multiscale feature extraction strategy to capture both detailed and contextual features from images, allowing for a more comprehensive analysis.

2. **Integration of Spatial-Reduction Attention**: Employ spatial-reduction attention mechanisms to enhance focus on key regions of the image, effectively reducing noise from less important areas.

3. **Progressive Dimensional Reduction**: Apply progressive dimensional reduction techniques to maintain a hierarchy of features, ensuring that crucial information is preserved while minimizing dimensional complexity.

4 Dataset

The dataset used is the *ArASL Database Grayscale* [5], which contains grayscale images of Arabic sign language gestures. Each image represents a unique hand gesture corresponding to an Arabic alphabet letter. The dataset was and divided into training (shuffled), validation, and test sets, and images were resized to 224×224 pixels, converted to tensors, and normalized.

5 Implementation Details

5.1 Data Preparation

5.1.1 Data Loading and Transformation

We used the `datasets` library to load the ArASL Database Grayscale dataset, specifically chosen for Arabic sign language recognition due to its challenging grayscale images. The dataset was split into training, validation, and test sets. Each image was resized to 224×224 pixels, converted to tensors, and normalized to a standard distribution for improved model performance.

5.1.2 Custom Dataset Class

To handle the specific structure of the ArASL dataset, we developed a custom dataset class. This class efficiently handles image preprocessing, including conversion to grayscale and mapping of labels for each sample. The class enables seamless integration with PyTorch's `DataLoader`, supporting batched data loading for optimized model training and evaluation.

5.2 Model Architecture

Our primary goal was to implement a custom model architecture inspired by the methods in the main paper and the supplementary work. This model combines the power of a CNN backbone with multiscale feature extraction through transformer stages. We also evaluated a baseline U-Net, fine-tuned standard models (ResNet-50, GoogleNet, Vision Transformer), and two versions of our custom model with and without pretrained weights.

Figure 1 provides a visual representation of our custom model architecture.

5.2.1 CNN Backbone: Adjusted ResNet-18 for Grayscale Images

We employed ResNet-18 as the backbone for our custom model, creating two variations to explore the impact of transfer learning. In both cases, we adjusted the first convolutional layer to accept a single input channel instead of three, enabling effective processing of grayscale images.

The first version of our backbone was initialized with pretrained weights from ImageNet, leveraging transfer learning to provide a strong feature extractor. This pretrained model offered the advantage of prior knowledge, potentially improving convergence speed and final accuracy.

For the second version, we implemented a custom ResNet-18 from scratch without using pretrained weights. This model allowed us to assess the performance of our architecture without the benefits of transfer learning, offering insights into the model's learning capability when trained solely on our specific dataset.

5.2.2 Multiscale Transformer Network

The multiscale transformer network was implemented with progressive dimension reduction across transformer stages, inspired by the MVTN approach from the supplementary paper. Each transformer stage comprises multi-head self-attention and feed-forward layers, progressively reducing the embedding dimension to capture hierarchical features at multiple scales.

Transformer Stages Each transformer stage follows a specific design to capture multiscale features:

- **Self-Attention Mechanism**: Attention layers compute the importance of different image regions, capturing local and global context.
- **Dimension Reduction**: A linear projection reduces feature dimensionality between stages, enabling efficient processing of multiscale information without losing essential features.

Model Initialization and Classification Head The multiscale transformer network outputs features passed through a global average pooling layer, followed by a classification head. The classification head consists of a layer normalization and a fully connected layer, outputting predictions for each of the 32 sign language classes.

5.2.3 Training Parameters

The experiments were conducted under the following setup:

- **Loss Function:** Cross-Entropy Loss.
- **Optimizer:** Adam optimizer with a learning rate of 1×10^{-4}.
- **Batch Size:** 32.
- **Number of Epochs:** 1.
- **GPU:** A100.

5.2.4 Forward Pass and Transformer Stages

The forward pass of our model consists of three main stages that progressively build and refine the feature representations:

1. **Stage 1 - Convolutional Feature Extraction:** After preprocessing, the input image (224x224x1) enters the CNN backbone, where it goes through a 7x7 convolution layer with 64 channels, followed by max pooling, resulting in a reduced size of 56x56x512.

2. **Stage 2 - Transformer Layer for Feature Contextualization:** The feature maps from the CNN are then processed by a three-stage transformer module. Each stage applies multi-head self-attention to capture spatial dependencies, followed by an MLP for further transformation, with each stage progressively reducing the embedding dimension.
   - *Stage 1 Transformer Output:* Produces a feature tensor of size 56x56x512.
   - *Stage 2 Transformer Output:* Reduces feature depth to 256 channels, maintaining spatial dimensions.
   - *Stage 3 Transformer Output:* Reduces feature depth to 128 channels, preserving essential information in a compact representation.
3. **Stage 3 - Classification Head:** The final output from the transformer stages undergoes global average pooling, followed by a fully connected layer that maps to the 32 class probabilities, providing the final classification output.

5.3 Mathematical Explanation

5.3.1 Self-Attention Mechanism

The self-attention mechanism computes the output as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1}$$

where:

- Q is the query matrix.
- K is the key matrix.
- V is the value matrix.
- d_k is the dimension of the key vectors.

In this self-attention setup, the Q, K, and V matrices are identical since they originate from the same source input.

5.3.2 Multi-Head Attention

Multi-head attention extends self-attention by allowing the model to attend to different parts of the input from multiple perspectives. It is computed as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2}$$

where each head i is calculated as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

This approach enables the model to capture varied contextual information, improving accuracy in complex image recognition tasks.

5.3.3 Dimension Reduction

To optimize computational efficiency and feature representation, each transformer stage progressively reduces the feature dimension:

$$x_{\text{out}} = x_{\text{in}} W_{\text{proj}} \tag{4}$$

where $W_{\text{proj}} \in R^{d_{\text{in}} \times d_{\text{out}}}$ is a learnable projection matrix that transforms the input to a lower-dimensional space. This reduction maintains the model's computational efficiency without sacrificing essential details.

6 Running Instructions

6.1 Model Training Instructions(Optional)

After installing the necessary dependencies listed in the requirements.txt file, the model-building process should run seamlessly without requiring any additional configuration steps. This step is optional, as we have provided pre-trained models within the data folder for immediate use. The dataset is automatically downloaded via the Hugging Face datasets library, ensuring a smooth and efficient setup for your convenience.

6.2 Using the Model for Classification

We've developed an intuitive UI that allows users to seamlessly leverage our model for sign language recognition, enhancing visualization and simplifying the overall experience by displaying the image used and the generic image of the predicted letter in Arabic sign language. To ensure smooth functionality, essential functions required for the model's operation are pre-defined within the main.py file, which can be found in the src folder. In order for the script to work, please ensure the saved model file(custom_model.pth) is located in the same directory as main.py. In addition, please ensure that LabelMapping.txt, which can be found in the data folder, is also in the same directory as main.py.

7 Conclusion on Implementation

This study demonstrated the effectiveness of progressively deeper and more complex architectures for Arabic sign language recognition, especially with pretrained models. The U-Net's limited performance highlighted the need for feature extraction depth, which we addressed by using pretrained ResNet-50, GoogleNet, and Vision Transformer models. Our custom model with multiscale transformer stages and pretrained ResNet-18 yielded the most promising results. The comparison between pretrained and non-pretrained versions of the ResNet-18 feature extractor further emphasized transfer learning's value. These findings will guide future research in leveraging multiscale transformers and pretrained backbones for specialized visual recognition tasks.

Figure 1 provides a visual representation of our custom model architecture.

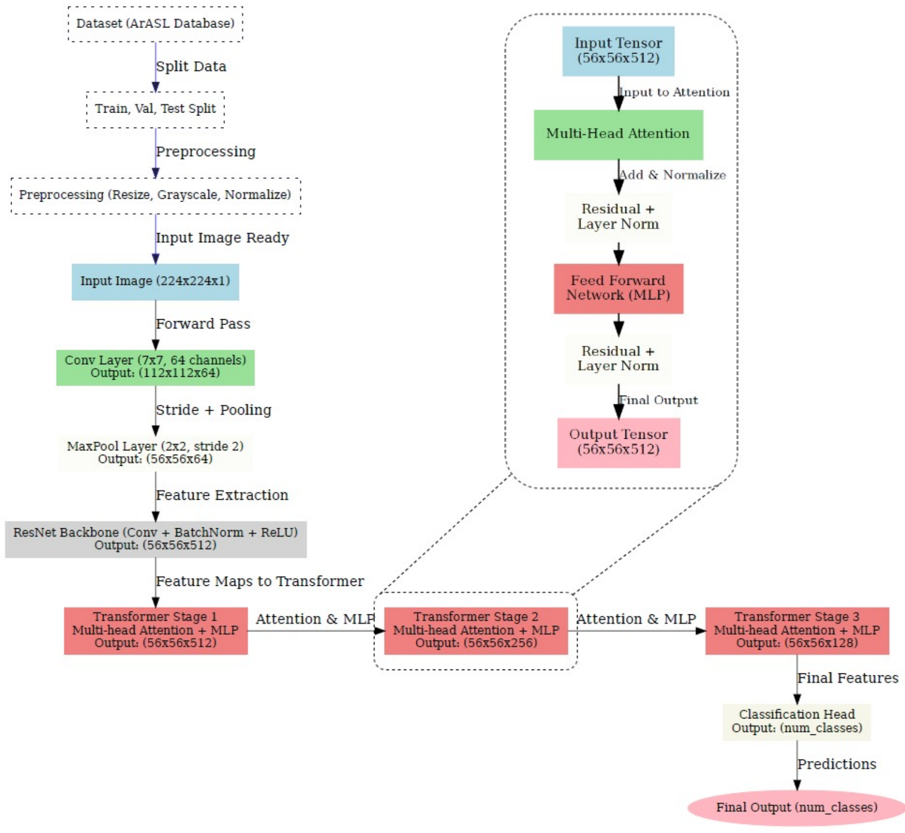


Figure 1: Architecture of Custom Model with Multiscale Feature Extraction and Attention Mechanisms

7.1 Challenges and Solutions

During implementation, we encountered several challenges:

7.1.1 Low Accuracy with U-Net

The initial U-Net model yielded low accuracy close to random guessing (3.5%). This issue was attributed to the U-Net's inability to capture complex features necessary for Arabic sign language recognition. To address this, we shifted to deeper models with pretrained weights.

7.1.2 Limited Performance and Training Time with ResNet-50

While training ResNet-50, we observed that although it performed better than the initial U-Net model, it still did not achieve satisfactory accuracy for our task. Additionally, training ResNet-50 for 60 epochs proved to be time-consuming, making it less practical for rapid experimentation and tuning. These limitations led us to explore other models that could provide higher accuracy with improved training efficiency.

7.1.3 Handling Fast Convergence with GoogleNet and ViT

The GoogleNet and Vision Transformer models achieved high accuracy within the first epoch. To visualize training progress accurately, we plotted accuracy and loss curves across batch increments, highlighting the rapid convergence while ensuring meaningful insights from the training process.

7.1.4 Computational Complexity

In our custom model, balancing feature richness and computational efficiency was challenging. The progressive dimensional reduction was critical in reducing complexity without losing significant features, making the model feasible for real-time applications.

7.1.5 Comparing Pretrained and Non-Pretrained Models

We trained two versions of the custom model, one with pretrained ResNet-18 and one without, to evaluate the impact of transfer learning. Results indicated that pretrained weights provided a beneficial starting point, leading to faster convergence and higher accuracy. This insight underscores the importance of transfer learning in image recognition tasks, especially with smaller or specialized datasets.

8 Results and Discussion

The various models demonstrate a step-by-step improvement in accuracy and robustness, validating our decision to progress from a basic U-Net to pretrained, deeper architectures. The custom model's integration of multiscale feature extraction and attention mechanisms, inspired by the referenced papers, provided the most promising results for Arabic sign language recognition. The performance differences between pretrained and non-pretrained ResNet-18 also highlight transfer learning's role in accelerating model convergence and improving accuracy.

8.1 Training Summary

Table 1: Model Training Comparison

Model	**Validation Loss**	**Validation Accuracy**	**Epochs**	**Time (mins)**
U-Net	3.4601	∼3.95%	10	∼5
ResNet-50	0.7060	∼82.50%	10	∼8
GoogleNet	0.0414	∼99.09%	1	∼1
Vision Transformer (ViT)	0.0881	∼97.56%	1	∼5
Custom Model (Pretrained)	0.1008	∼97.95%	1	∼1
Custom Model (Non-Pretrained)	0.3169	∼91.45%	1	∼1

8.2 Performance Summary

Table 2: Model Testing Comparison

Model	**Test Loss**	**Test Accuracy**	**F1 Score**	**F2 Score**	**Precision**	**Recall**	**AUC**
U-Net	3.4595	4.10%	0.0025	0.0055	0.0013	0.0312	0.5000
ResNet-50	0.7056	82.75%	0.8286	0.8256	0.8474	0.8267	0.9916
GoogleNet	0.0492	98.99%	0.9900	0.9900	0.9899	0.9901	0.9995
Vision Transformer (ViT)	0.1022	96.98%	0.9701	0.9699	0.9712	0.9700	0.9996
Custom Model (Pretrained)	0.1119	97.65%	0.9773	0.9769	0.9787	0.9767	0.9991
Custom Model (Non-Pretrained)	0.3272	91.14%	0.9034	0.9032	0.9354	0.9069	0.9974

8.3 Comparison Summary

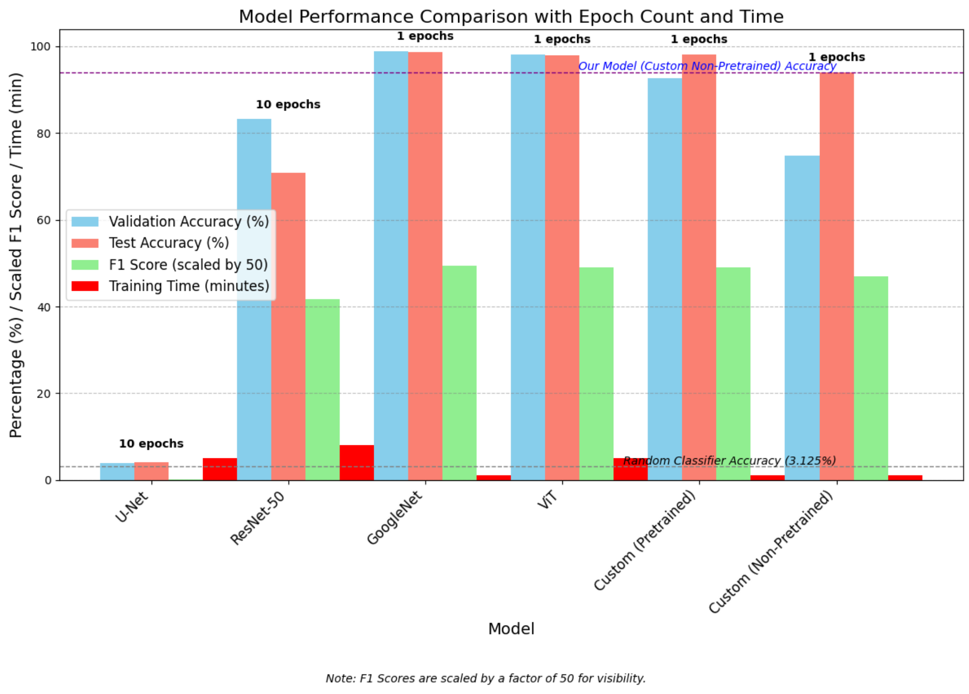
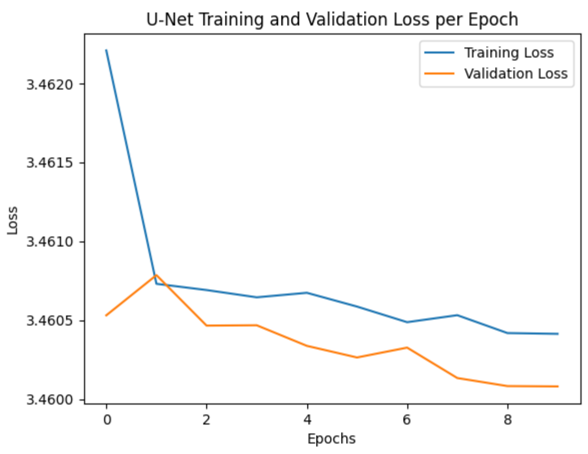
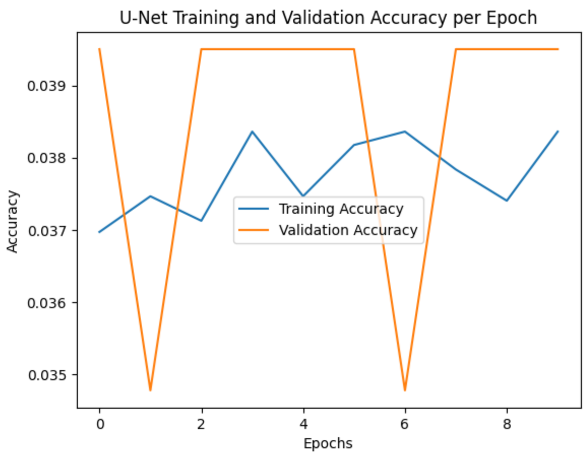


Figure 2: Comparison between all models

8.4 Training and Validation Curves

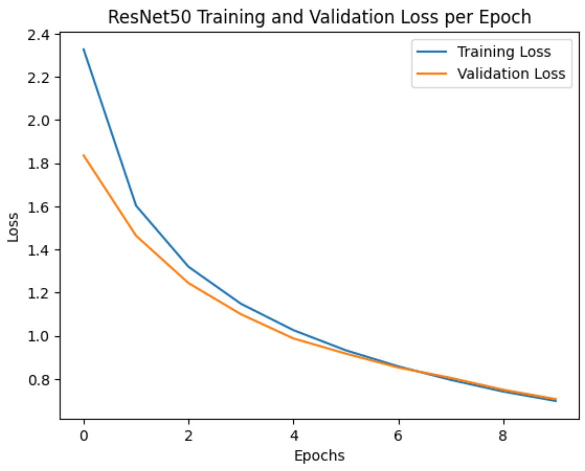


(a) Training and Validation Loss for U-Net

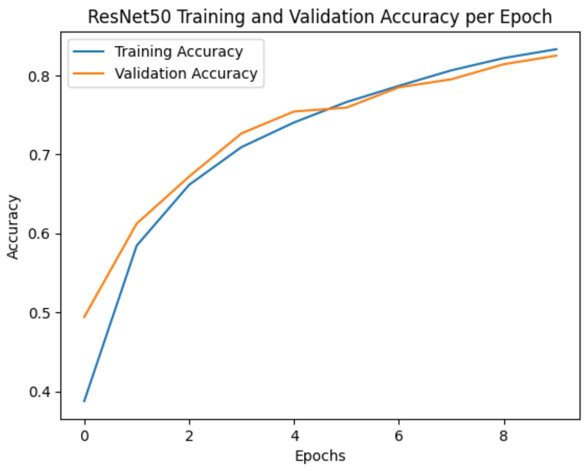


(b) Training and Validation Accuracy for U-Net

Figure 3: Performance Metrics for U-Net Model



(a) Training and Validation Loss for ResNet-50



(b) Training and Validation Accuracy for ResNet-50

Figure 4: Performance Metrics for ResNet-50 Model

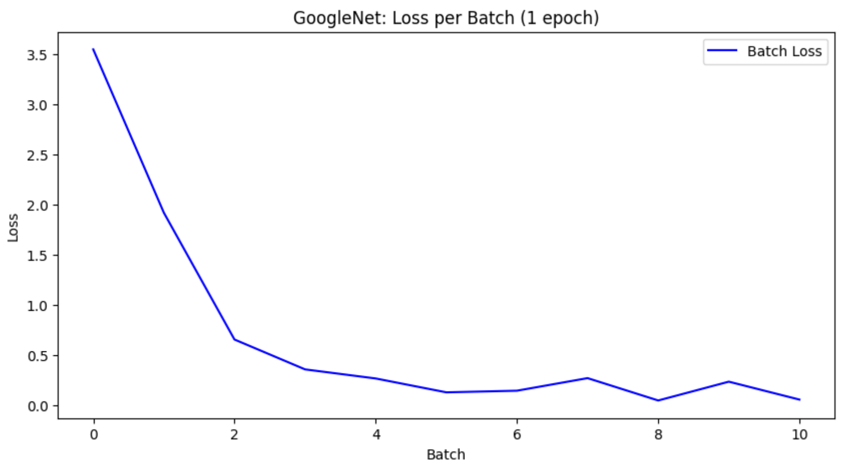


Figure 5: Training Loss for GoogleNet

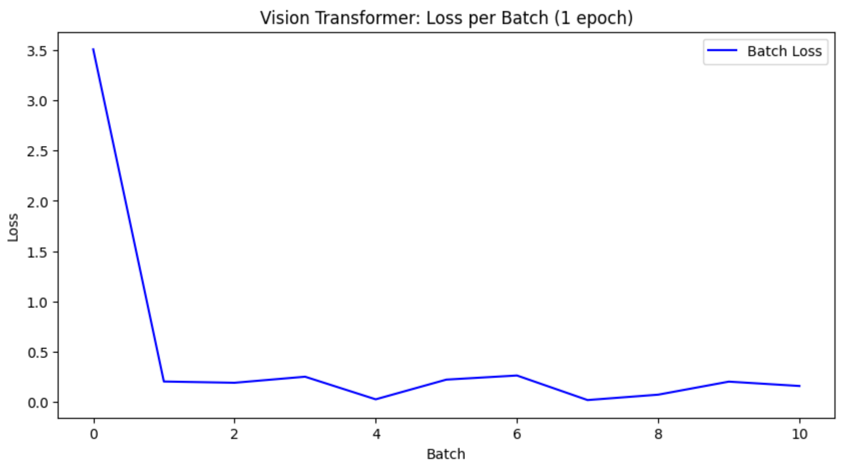


Figure 6: Training Loss for Vision Transformer (ViT)

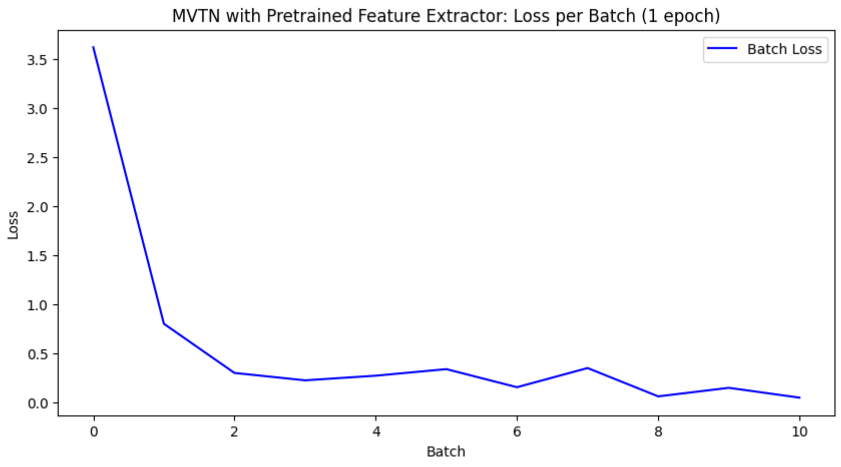


Figure 7: Training Loss for Custom Model (Pretrained backbone)

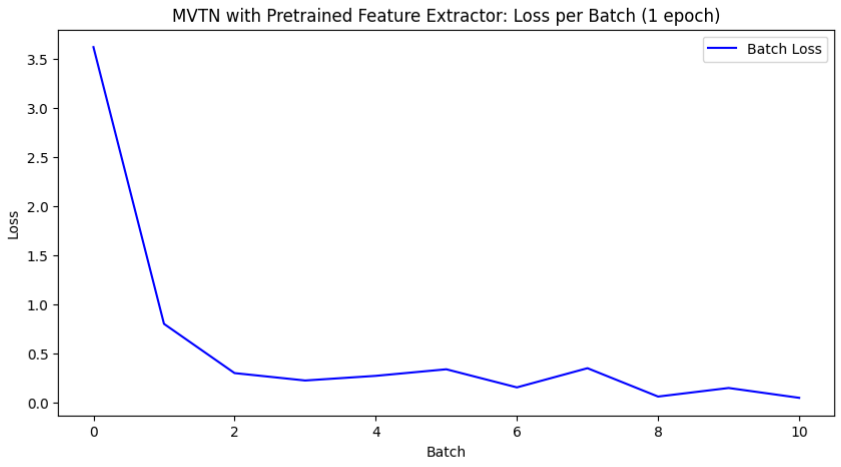


Figure 8: Training Loss for Custom Model (Non-Pretrained backbone)

8.5 Discussion of Results

8.5.1 Basic U-Net

U-Net achieved poor results with a validation accuracy of 3.95% and test accuracy of 4.10%. Its simple architecture couldn't capture the complexity of Arabic Sign Language gestures, demonstrating the need for more advanced models.

8.5.2 Fine-Tuned ResNet-50

ResNet-50 showed significant improvement with a validation accuracy of 82.5% and test accuracy of 82.75%. Its deeper architecture and residual connections allowed it to learn more complex features but still fell short of top-performing models.

8.5.3 Fine-Tuned GoogleNet

GoogleNet achieved impressive performance, with a validation accuracy of 99.09% and test accuracy of 98.99%. The inception modules allowed it to efficiently capture multi-scale features, leading to rapid convergence and excellent results.

8.5.4 Fine-Tuned Vision Transformer (ViT)

ViT reached a validation accuracy of 97.56% and test accuracy of 96.98%. The self-attention mechanism allowed it to capture intricate dependencies across the image, providing strong performance, though slightly behind GoogleNet.

8.5.5 Custom Model with Pretrained ResNet-18

This model achieved a validation accuracy of 97.95% and test accuracy of 97.65%. Pretraining on a larger dataset helped the model converge quickly and perform well by transferring learned features.

8.5.6 Custom Model with Non-Pretrained ResNet-18

The non-pretrained model achieved a validation accuracy of 91.45% and test accuracy of 91.14%. It performed well but lagged behind due to the absence of pretrained weights, which hindered feature learning and convergence. Despite this, the model still achieved a solid F1 score of 0.9034 and AUC of 0.9974, indicating strong performance despite its limitations.

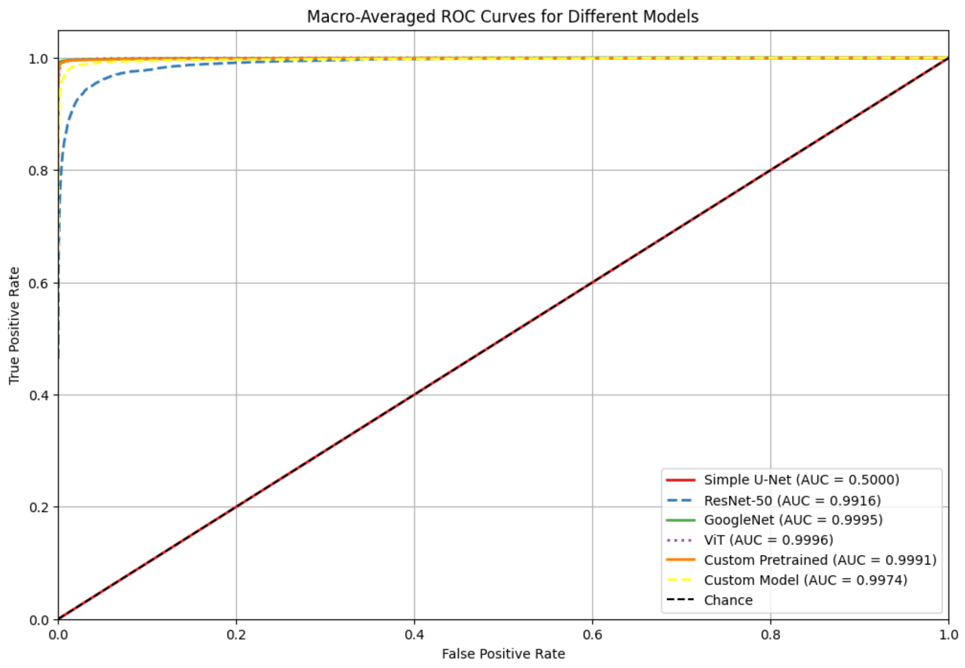


Figure 9: AUC Comparison

9 Conclusion

In this study, we developed an enhanced framework for Arabic Sign Language recognition by leveraging multiscale feature extraction, spatial-reduction attention, and progressive dimensional reduction. Our results demonstrate that advanced architectures like ResNet-50, GoogleNet, and Vision Transformer can significantly outperform simpler models like U-Net in capturing the complex features required for accurate gesture recognition. The custom model with a pretrained ResNet-18 backbone further highlighted the benefits of transfer learning, achieving high accuracy and efficient convergence.

The proposed framework shows promising potential for improving image recognition systems in specialized applications like sign language interpretation. By effectively capturing both detailed and contextual features, our approach can serve as a foundation for future research in non-verbal communication recognition. Our findings underscore the importance of utilizing deep, multiscale architectures and transfer learning to achieve high performance in image analysis tasks, especially in scenarios with limited data.

Future work could explore further optimizations, such as fine-tuning the model on a larger dataset or integrating additional attention mechanisms to enhance performance and expand the framework's applicability to other complex visual recognition challenges.

References

[1] Vaswani, A., et al. (2017). Attention is All You Need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008.

[2] Dosovitskiy, A., et al. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929.*

[3] Garg, M., Ghosh, D., Pradhan, P. M. (2024). MVTN: A Multiscale Video Transformer Network for Hand Gesture Recognition. *arXiv preprint arXiv:2409.03890.*

[4] Balat, M., et al. (2024). Advanced Arabic Alphabet Sign Language Recognition Using Transfer Learning and Transformer Models. Egypt-Japanese University of Science & Technology. *arXiv preprint arXiv:2410.00681.*

[5] ArASL Database. `https://huggingface.co/datasets/pain/ArASL_Database_Grayscale`