COSE474-2024F: Final Project Report
"Enhancing Meta-Network Architectures in CoCoOp"

Nayoung Kim [1]

1. Introduction

1.1. Motivation

Recently, pre-trained vision-language models like CLIP (Contrastive Language-Image Pre-training) have emerged as powerful tools in computer vision, demonstrating exceptional performance across various tasks through their ability to connect visual and textual information. These models leverage massive datasets of image-text pairs to learn robust visual representations that can generalize across different domains and tasks.

To enhance these models' performance, several prompt learning methods have been proposed. To enhance these models' performance, several prompt learning methods have been proposed. Context Optimization (CoOp) introduced learnable continuous prompts to replace hand-crafted text prompts, showing improved performance on various downstream tasks. Building upon this, Conditional Context Optimization (CoCoOp) incorporated instance-conditional prompts to better handle visual variations. Prompt-based Meta-learning with Residual Adaption (ProMetaR) further advanced the field with meta-learning strategies. Besides, these models have been facing challenges in adapting to new domains while maintaining performance on base classes.

This work focuses on enhancing CoCoOp's meta-network through architectural improvements the meta-network, which generates context-aware prompts. The original implementation relies on a relatively simple architecture composed of basic linear layers and activations, and I wanted to check if this is potentially limiting the model's capability to capture more complex feature relationships and generalize more effectively. Therefore, I focused on the CoCoOp's meta network and tried to achieve architectural improvements to enhance both base class performance and generalization to new classes.

1.2. Problem definition & challenges

While CoCoOp effectively learns prompts for visual recognition, its simple meta-network architecture may limit its ability to capture complex feature relationships and generalize to unseen classes. This work first focuses on understanding and implementing CoCoOp model and then addressing this limitation by exploring and evaluating different architectural modifications to enhance the performance aspects.

1.3. concise description of contribution

I propose and evaluate three architectural variants of CoOp's meta-network: First is a deeper architecture for enhanced feature extraction. Second is a residual architecture for improved gradient flow and feature preservation. Third is an attention-based architecture for dynamic feature relationship learning.

2. Methods

Park et al. (2024) have implemented basic structures of CoCoOp based on CLIP which I used in this study. However, the previous study did not focused on enhancing meta network architecture. This work aims to find better architectures of meta network for CoCoOp rather than composition of linear layer, relu activation, and linear layer as the main challenge. For that I implemented 3 variants of meta network architecture, which were mentioned above as deeper, residual, and attention-based architecture.

The original meta-network employs a straightforward structure with two linear layers connected by a ReLU activation function. This simple architecture serves as our baseline for comparison.

The 3 variants of meta network architecture is as follows.

First, deeper architecture was implemented with the addition of layers and ReLU activations. The deeper architecture expands upon this by implementing four sequential layers, and each intermediate layer is equipped with ReLU activation, creating a deeper transformation path for feature processing.

Second, the residual architecture introduces a skip connection mechanism while maintaining a moderate depth of

[*] Equal contribution [1] Department of Media & Communication, Double Major: Department of Computer Science, Korea University, Seoul, Korea. Correspondence to: Nayoung Kim <tkfkd9008a@korea.ac.kr>.

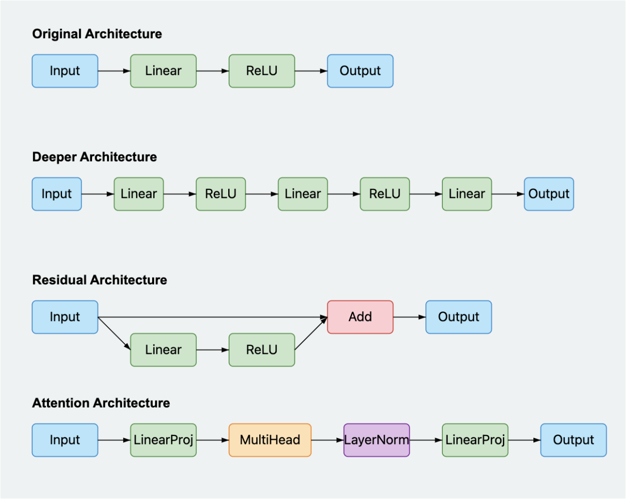


Figure 1. 4 Network Architectures

three layers. Residual connections preserve feature information, allowing the network to learn residual features while preserving direct access to input information.

Third, attention-based architecture was implemented with multi-head attention mechanism. It first projects the input features to a lower dimension and then processes them through an 8-head attention mechanism. Each head computes query, key, and value projections, enabling the network to capture different aspects of feature relationships. The attention outputs are combined and processed through a final two-layer projection network with ReLU activation to reach the context dimension. This architecture also includes dropout (0.1) for regularization. In this case feature relations was considered, based on the idea of processing static feature limits the contextual understanding.

Figure 1 demonstrates the main figure and structure of the algorithm.

3. Experiment

3.1. Dataset & Computing Resource

Our experimental results on the EuroSAT dataset, which is a sentinel-2 satellite image dataset consisting of 27,000 labeled images with 10 distinct land-use and land-cover classes. This data was considered suitable for this research because it has domain specificity and it provides a balanced distribution across classes, which allows for effective evaluation of both base and new class performance. In this experiments, I used 80 training samples for base classes(16 per class) and evaluated on both base class test set (4,200 samples) and new class test set (3,900 samples). This setup was expected to effectively test the model's ability to learn from limited data while maintaining generalization capabilities.

The model and experiments are implemented using PyTorch and trained on a NVIDIA Tesla T4 GPU(16GB) provided by Google Colaboratory.

3.2. hyperparameters

All experiments were conducted with consistent hyperparameters across different architectures for fair comparison. The key hyperparameters are as follows:

Table 1. Dataset and Training Configuration

Parameter	Value
Base classes	5
New classes	5
Training samples per class	16
Total training samples	80
Validation samples	20
Base class test samples	4,200
New class test samples	3,900
Batch size	4
Number of epochs	100

3.3. Architecture Specifications

For all architectural variants, we maintained consistent dimensionality:

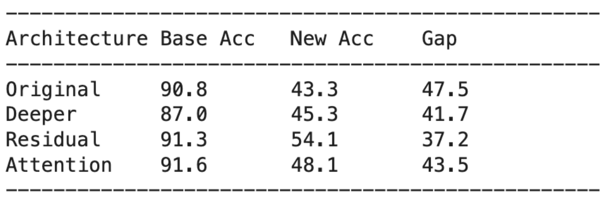
- Input dimension (d_{vis}): CLIP ViT-B/16 output dimension
- Hidden dimension (d_h): $d_{vis}/16$
- Output dimension (d_{ctx}): CLIP text encoder dimension

Architecture-specific parameters:

Table 2. Architecture-Specific Parameters

Architecture	Parameter	Value
Deeper	Number of layers	4
	Activation	ReLU (inplace=True)
Residual	Main branch layers	3
	Activation	ReLU (inplace=True)
Attention	Attention heads	8
	Dropout rate	0.1
	Scale factor	$(d_h/num\_heads)^{-0.5}$
	Q, K, V dimensions	d_h

All implementations support FP16 precision training when specified in the configuration. The learning rate and optimization parameters follow the original CoCoOp settings to maintain consistency with the baseline.



```
--------------------------------------------------
Architecture Base Acc   New Acc    Gap
--------------------------------------------------
Original     90.8       43.3       47.5
Deeper       87.0       45.3       41.7
Residual     91.3       54.1       37.2
Attention    91.6       48.1       43.5
--------------------------------------------------
```

Figure 2. Numerical Results

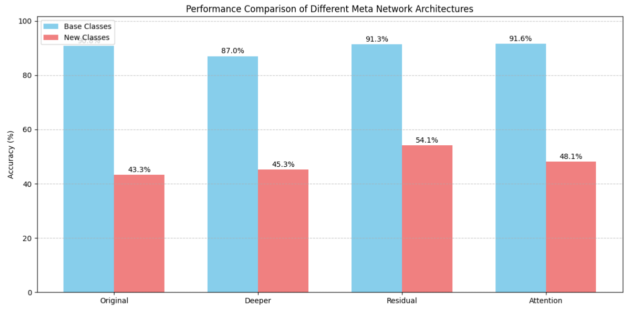


Figure 3. performance comparison of different meta network architectures

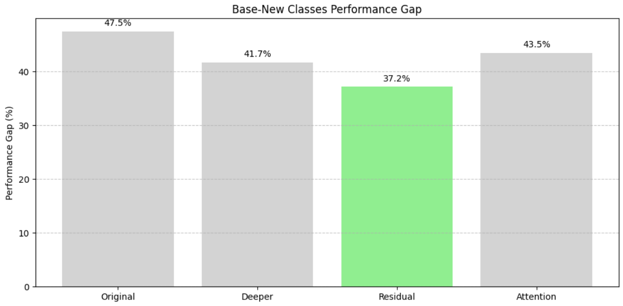


Figure 4. Base-New performance gap

3.4. Results

As a result of the experiments, distinct performance patterns across different architectural variants was observed. The attention-based architecture achieved the highest accuracy of 91.6%, showing a little higher performance over the residual architecture which achieved 91.3%. On the other hand, deeper architecture underperformed on 90.8% of base classes, achieving only 87.0% accuracy.

For new class generalization, the residual architecture demonstrated superior performance, achieving 54.1% accuracy, a significant improvement of 10.8 percentage points over the original architecture's 43.3%. The attention and deeper architectures showed moderate improvements in new class performance, reaching 48.1% and 45.3% respectively.

The experimental results reveal interesting patterns in how different architectural choices affect the model's behavior. The residual architecture's success in both base and new classes suggests that preserving original feature information while learning new transformations is crucial for balanced performance.

On the other hand, the attention mechanism's strong performance on base classes but relatively lower improvement on new classes indicates its effectiveness in learning specific feature relationships but potential limitations in generalization. This reminds the importance of considering the overfitting problem in deep learning.

Moreover, the performance gap between base and new classes can be added to the criteria of evaluation. The residual architecture demonstrates the smallest gap (37.2%), which is significantly lower than the original architecture's gap (47.5%). This reduced gap suggests that the residual connections effectively balance feature preservation and adaptation.

3.5. Discussion

The success of the residual architecture can be explained by its ability to maintain a direct path for original feature propagation while learning additional transformations. This architectural choice appears particularly beneficial in our limited-data scenario (80 training samples), where preserving core feature information is crucial.

The deeper architecture's unexpected underperformance suggests that simply increasing network depth may not be beneficial in prompt learning scenarios with limited data. Furthermore, the attention mechanism's performance pattern indicates its strength in learning specific feature relationships but potential limitations in generalizing to unseen classes due to the overfitting problem.

These findings suggest that architectural choices in meta-networks significantly impact both base class performance and generalization capability. The residual architecture's balanced performance makes it particularly suitable for real-world applications where both aspects are important. This study also can highlight an important trade-off between maintaining base class performance and improving generalization to new classes.

4. Limitations & Future directions

There are several limitations that should be addressed in future research.

While we achieved improvements with both base and new classes with the residual architecture, the persistent performance gap suggests that further architectural innovations might be needed to fully bridge this divide.

One compelling direction is the development of hybrid architectures that combine the strengths of different approaches. Integrating residual connections with attention mechanisms may be a way to leverage both feature preservation and dynamic relationship learning.

Exploring other combinations of architectures would also work. As this study explored three architectural variants but there could be other potential architectures or combinations that we haven't investigated.

I may also mention that the current design uses fixed hyperparameters (e.g., number of attention heads, layer dimensions) which might not be optimal for all scenarios. Attention-based architectures, in particular, may needed higher memory requirements, increased computational resources and longer training times, so there might existed some limits of their applications. Further studies may also consider this when designing new architecture.

5. Reference

1. Jinyoung Park, Juyeon Ko, Hyunwoo J. Kim, ”Prompt Learning via Meta-Regularization” (2024)

2. Radford et al., ”Learning Transferable Visual Models From Natural Language Supervision” (2021)

3. Zhou et al., ”Learning to Prompt for Vision-Language Models” (2021)