Land Cover Classification Using Foundation Models

Haiyang Jiang
haiyang.jiang@polytechnique.edu

Jingnan Cao
jingnan.cao@polytechnique.edu

Abstract

The study of Multimodal Large Language Models (MLLMs) has gained significant attention recently as a burgeoning research area. These models employ the potent abilities of Large Language Models (LLMs) to handle tasks that encompass various modalities, essentially functioning as a central processor and has shown promising results in representation and transfer learning. Among the various models, CLIP has consistently stood out as the most impactful. However, CLIP's performance is at its poorest when dealing with the EuroSAT dataset, which relies on texture-based image classification rather than object-based classification. In this particular project, we have implemented several few-shot techniques, including linear probing, which is mentioned in the CLIP paper, and additional state-of-the-art prompting engineering methods such as CoOp and CoCoOp, to enhance the model's performance on the EuroSAT dataset. Our repo can be seen at: https://github.com/iLori-Jiang/CLIP_on_EuroSAT

1. Introduction

Large-scale deep network models pretrained on ultra large-scale data on the internet, whether text or images have shown impressive performance recently. To apply these powerful models to leverage the representation they have learned to downstream tasks, there are several approaches. The easiest one is the zero-shot transfer without fine-tuning. While zero-shot transfer performs well on some datasets, it is generally better to adapt the model itself if there are any labeled examples available.

In the traditional representation learning task, knowledge is based mostly on discretized labels, which is the one-hot labels in the vision classification task and have no language meaning. On the other hand, newly emerged vision-language pre-training like CLIP [5], which is trained on over 400M pairs of image and text descriptions collected from the internet, aligns images and texts in a common feature space, which ensures the model understands the language meaning behind the image. Thus it allows zero-shot transfer to a downstream task via either prompting, which is by generating the classification weights from the natural language description of the class, or via linear probing, which directly adopt the image representation from the image encoder of the model and train a classifier head with the available labelled data on top of the representation.

For prompting engineering, it is one of the major challenges for deploying such models in practice since it requires domain expertise and is extremely time-consuming. One needs to spend a significant amount of time on words tuning since a slight change in wording could have a huge impact on performance. Inspired by recent advances in prompt learning research in natural language processing (NLP), Context Optimization (CoOp) [8] and Conditional Context Optimization (CoCoOp) [7] have been proposed specifically for adapting CLIP-like vision-language models for downstream image recognition. Concretely, CoOp and CoCoOp models a prompt's context words with learnable vectors while the entire pre-trained parameters are kept fixed. By this, we can obtain the optimal prompting for the specified tasks in an autonomous manner, instead of manually searching, which will improve the performance of the model on the downstream task.

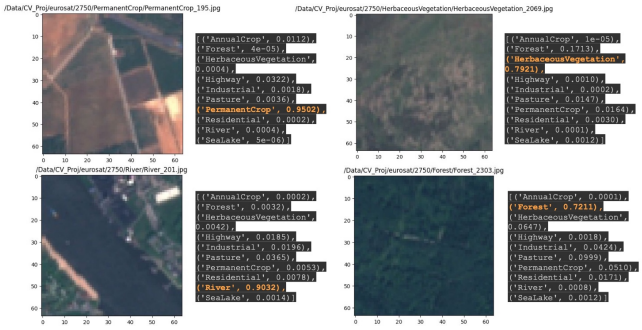


Figure 1. Some results of zero-shot inference using the CoOp learned unified context prompt.

What's more, according to the original CLIP paper, EuroSAT [3] remains a challenging dataset for CLIP's zero-performance, where it performs 37.1% worse than the fully supervised ResNet50 model. Thus, it's interesting to investigate which fine tunning approach can facilitate and improve the performance of CLIP on this dataset.

Therefore, in this project, we propose four candidates to compare and demonstrate the performance of each candidate approach on adapting the foundation model CLIP to land cover classification task using EuroSAT dataset. The candidate approaches are: zero-shot inference, fully representation linear probing, few-shot linear probing, and prompt learning.

2. Background

Pre-training methods that learn directly from raw text have revolutionized NLP recently[2]. Task-agnostic objectives such as autoregressive and masked language modeling have scaled across many orders of magnitude in compute, model capacity, and data, steadily improving capabilities. The development of "text-to-text" as a standardized input-output interface[6] has enabled task-agnostic architectures to zero-shot transfer to downstream datasets removing the need for specialized output heads or dataset-specific customization.

However, in other fields such as computer vision, it is still standard method to pre-tain models on crowd-labeled datasets such as ImageNet[1]. People begin to consider whether scalable pre-training methods that learn directly from web text could result in a similar breakthrough in computer vision. Enabled by the large amounts of publicly available data on the internet, the OpenAI creates a new dataset of 400 million (image, text) pairs and demonstrates that a simplified version of ConVIRT trained from scratch, for Contrastive Language-Image Pre-training, is an efficient method of learning from natural language supervision.

2.1. Contrastive Language-Image Pre-training (CLIP)

CLIP consists of two parallel encoders for processing images and text, which aim to map high-dimensional data into a low-dimensional embedding. Then the outputs of the encoders are projected into a shared embedding space. The text encoder is a Transformer, while the image encoder is a Vision Transformer (ViT) or a CNN-like ResNet (RN). For our experiments, we utilize the open-sourced pretrained CLIP models. Specifically, given a sequence of words (tokens), such as "a photo of a dog", CLIP first converts each one of the token into a lower-cased byte pair encoding (BPE) representation, which is essentially a unique numeric ID. The vocabulary size in CLIP is 49,152. Each text sequence is capped at a fixed length of 77. After that, the IDs are mapped to 512-D word embedding vectors, which are then passed on to the Transformer. Finally, the features at the last token position are layer normalized and further processed by a linear projection layer.

2.1.1 Training

CLIP is trained to align the two embedding spaces learned for images and text respectively. Specifically, the learning objective is formulated as a contrastive loss. Given a batch of image-text pairs, CLIP maximizes the cosine similarity for matched pairs while minimizes the cosine similarity for all other unmatched pairs. To learn diverse visual concepts that are more transferable to downstream tasks, CLIP's team collects a large training dataset consisting of 400 million image-text pairs.

CLIP further uses the similarity as prediction probabilities for classifying an image with the correct text caption (or vice versa) across batches. Formally, denote I and T as the set of image and text features in a single batch. The prediction probability for the i^{th} image and j^{th} caption in the batch is given by

$$p(T_j|I_i) = \frac{\exp(\cos(T_j, I_i)/\tau)}{\sum_{T_k \in T} \exp(\cos(T_k, I_i)/\tau)} \tag{1}$$

where τ is a learnable temperature parameter.

2.1.2 Inference

For a downstream classification task at test time, CLIP first embeds the textual descriptions of all classes. These descriptions can be a phrase like "a photo of a <class>", or the heavily and manually engineered embeddings ensembled over 80 different templates as showed in the CLIP paper. Each image is then classified based on the similarity of its image embedding and classes embedding to determine the most-likely class.

2.1.3 Advantage against traditional vision system

Traditional vision system such as the classifier learning approach are focusing on closed-set visual concepts, which consists of one-hot no meaningful label. Therefore, the representation learned by the model is only related to the fix numerical label. As we want to extend its knowledge to new unseen label, the model requires fully retrain, which can be strongly time-consuming. On the other hand, vision-language pre-training allows open-set visual concepts to be explored through a high-capacity text encoder, leading to a broader semantic space and in turn making the learned representations more transferable to downstream tasks.

What's more, the output of the traditional vision system is based on linear classifier head, where the number of the channel is fixed. Therefore, the output is static and lack of dynamic. Adopting vision-language pre-training, the output has no limitation and thus strongly dynamic, which can be suitable for any potential task design.

3. Methods

Although zero-shot CLIP performs well on natural images and general object classification datasets, its performance drops quickly on more abstract tasks from out-of-distribution data. Substantial gains can be achieved by fine-tuning the pre-trained model with the following approaches:

3.1. Linear Probe

A classic method of fine tuning the large foundation model is to train a linear probe on top of frozen features. Given a pre-trained CLIP model, we discard the text encoder, freeze the image encoder, and learn a linear layer on top of the image features before they're projected to the shared embedding space. The linear layer maps the image features to logits of the class predictions. While this simple method is popular and effective, it's parameter-inefficient for tasks with higher number of classes and fails to leverage any of the language information contained in CLIP.

3.2. Prompt Learning

Alternatively, we can consider adding parameters which act on the model input. Such an approach known as prompt learning has emerged as a parameter-efficient fine-tuning method in language. A fixed number of continuous vectors (a "prompt") is provided to the model input and optimized throughout training. Prompt tuning is parameter-efficient and removes the need for manual prompt engineering. Ideally, the learned prompts would contain such domain-specific information. However, prompt tuning suffers from high variance during training and is sensitive to initialization.

3.2.1 Context Optimization (CoOp)

By CoOp, we can avoids manual prompt tuning by modeling context words with continuous vectors that are end-to-end learned from data while the massive pre-trained parameters are frozen. This approach is demonstrated to be requiring as few as one or two shots to beat hand-crafted prompts with a decent margin and is able to gain significant improvements over prompt engineering with more shots. For example, with 16 shots the average gain is around 15%. Despite being a learning- based approach, CoOp achieves superb domain generalization performance compared with the zero-shot model using hand-crafted prompts.

Given a limited set of labelled data on the downstream task, we could compute the similarity between the encoded query sentence, which is the combination of the prompt and the labelled class, and the encoded image. Therefore, for a learnable prompt, our objective is to maximize its similarity with the correct class. This is a simple classification problem and a cross entropy loss would be sufficient for this task. The gradient of the loss will backpropogate all the way through the frozen text encoder to the input prompt, and thus the prompt can be updated. The design of continuous representations also allows full exploration in the word embedding space, which facilitates the learning of task-relevant context.

1. Unified Context: the first is the unified context version, which shares the same context with all classes. Specifically, the prompt given to the text encoder is designed with the following form: $t = [V]_1[V]_2 \dots [V]_M[\texttt{CLASS}]$, where each $[V]_m$ ($m \in \{1, \dots, M\}$) is a vector with the same dimension as word embeddings (i.e., 512 for CLIP), and M is a hyperparameter specifying the number of context tokens.
2. Class-Specific Context: Another option is to design class-specific context (CSC) where context vectors are independent to each class, i.e., $[V]_1^i[V]_2^i \dots [V]_M^i \neq [V]_1^j[V]_2^j \dots [V]_M^j$ for $i \neq j$ and $i, j \in \{1, \dots, K\}$. As an alternative to unified context, the original paper finds that CSC is particularly useful for some fine-grained classification tasks.

3.2.2 Conditional Context Optimization (CoCoOp)

A critical problem of CoOp is that the learned context is not generalizable to wider unseen classes within the same dataset, suggesting that CoOp overfits base classes observed during training. To address the problem, the authors argue that instance-conditional context can generalize better because it shifts the focus away from a specific set of classes to each input instance. Therefore, they continue proposing CoCoOp, which extends CoOp by further learning a lightweight neural network to generate for each image an input-conditional token (vector). Compared to CoOp's static prompts, the new dynamic prompts adapt to each instance and are thus less sensitive to class shift. Extensive experiments show that CoCoOp generalizes much better than CoOp to unseen classes, even showing promising transferability beyond a single dataset; and yields stronger domain generalization performance as well.

For the implementation, it learn a lightweight neural network, called Meta-Net, to generate for each input a conditional token (vector), which is then added to the context vectors. Let $h_\theta()$ denote the Meta-Net parameterized by θ, each context token is now obtained by $v_m(x) = v_m + \pi$ where $\pi = h_\theta(x)$ and $m \in \{1, 2, \dots, M\}$.The prompt for the i^th class is thus conditioned on the input: $t_i(x) = \{v_1(x), v_2(x), \dots, v_M(x), c_i\}$.

During training, we update the context vectors $\{v_m\}_{m=1}^M$ together with the Meta-Net's parameters θ. In this work, the Meta-Net is built with a two-layer bottleneck structure (Linear-ReLU-Linear), with the hidden layer reducing the input dimension by $16\times$. The input to the Meta-Net is simply the output features produced by the image encoder.

4. Experiments

4.1. Dataset

An challenge emerged in the computer vision domain is the land use and land cover classification. The EuroSAT dataset is a collection based on Sentinel-2 satellite images, utilized for land use and land cover classification. It covers 13 spectral bands and comprises 10 classes with a total of 27,000 labeled and geo-referenced images. The classes include 'AnnualCrop', 'Forest', 'HerbaceousVegetation', 'Highway', 'Industrial', 'Pasture', 'PermanentCrop', 'Residential', 'River', and 'SeaLake'.

However, CLIP's zero-shot performance is bad on this dataset according to the original paper. This might be caused by (1) Domain Mismatch: The EuroSAT dataset consists of satellite images, which might significantly differ from the data CLIP was trained on. The domain-specific features present in satellite imagery may not be well-represented in CLIP's training data. (2) Abstract Concepts and Fine-grained Details: Land cover classification often requires understanding abstract concepts and fine-grained details based on subtle features, which is limited for zero-shot inference.

4.2. Training & Testing Details

The optimizer are both Adam with learning rate $1e^{-3}$ to $2e^{-3}$. The loss are both classic cross entropy loss for classification task. (1) For 16-shots linear probe, we only used 16 images per class for training and the whole dataset for testing, after 100 epochs the loss was reduced to 0.778. (2) As for representation linear probing on the whole dataset, we split the dataset with 80% for training and 20% for validation, after 100 epochs the loss reduced to 0.122. (3) For prompt learning with CoOp, we follow their official pipeline, with ResNet50 as the backbone, train for 200 epochs to learn the 16 context token vectors. The loss is reported on a validation set of 4 samples per class. The unified context achieved the final loss of 0.3680, while the class-speicfic context achieved 0.1410. (4) For prompt learning with CoCoOp, we follow their official pipeline, with ViT-B/16 as backbone, only train on 5 classes (half of the whole classes), train for 10 epochs to learn the 4 context token vectors and a Meta-Net. The loss is reported also on a validation set of 4 samples per class of the 5 training classes. The loss finally achieves 0.0394. Note that since the validation set is extremely small, the loss might vary a lot from epoch to epoch.

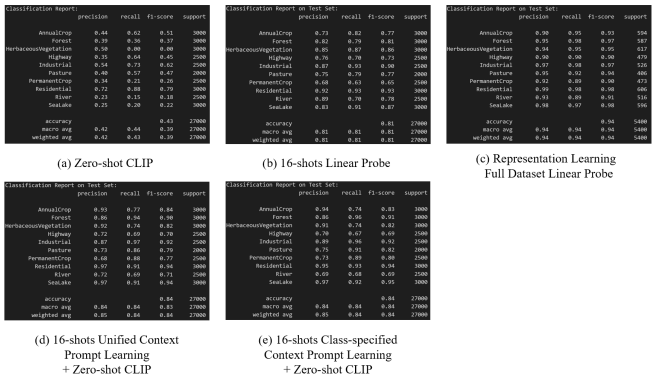
In the end, to provide a unified benchmark for all the methods, we test their performance again on all the dataset. For testing the generalization ability of CoCoOp on the unseen classes in the same dataset, this model is tested on all the data in the remain 5 classes.

5. Result

In our study, we evaluated various methods on the EuroSAT dataset. To assess their performance, we utilized classification reports and analyzed the results using confusion matrices.

As depicted in Fig. 2 and Fig. 3, the zero-shot CLIP model served as our baseline, showing the lowest accuracy overall. This underscores a substantial potential for improvement, particularly in the 'SeaLake' and 'River' categories. While representation linear probing can seen as the upper bound since it utilise all the dataset to learn the classifier head. The few-shots approaches can be considered as in the middle of zero-shot and fully representation learning.

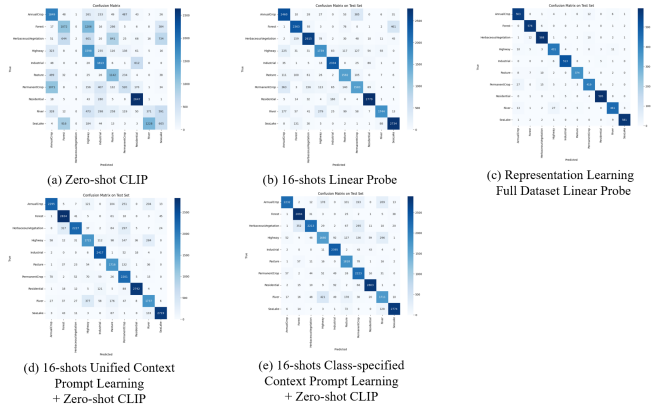
Besides, the CoCoOp adopt also 16-shots training, achieving the overall F1-score as 65% on the totally unseen data and classes, which is impressive. Given time limitation, we are failed to provide the detailed confusion matrix of this approach.



(a) Zero-shot CLIP (b) 16-shots Linear Probe (c) Representation Learning Full Dataset Linear Probe

(d) 16-shots Unified Context Prompt Learning + Zero-shot CLIP (e) 16-shots Class-specified Context Prompt Learning + Zero-shot CLIP

Figure 2. Classification report on the detailed classes



(a) Zero-shot CLIP (b) 16-shots Linear Probe (c) Representation Learning Full Dataset Linear Probe

(d) 16-shots Unified Context Prompt Learning + Zero-shot CLIP (e) 16-shots Class-specified Context Prompt Learning + Zero-shot CLIP

Figure 3. Confusion matrix on the detailed classes

What's more, since we have learnt the optimal prompt, we try to interpret them. It is difficult because the context vectors are optimized in a continuous space. We try

Figure 4. Interpretation on the learned unified context prompts, with each position we select top 5 closest neighbors. The distance is shown as the number on the right

to search within the vocabulary for words that are closest to the learned vectors based on the Euclidean distance. Fig. 4 shows the searched results. We observe that a few words are somewhat relevant to the tasks, which have been marked in red. But when connecting all the nearest words together, the prompts do not make much sense. Overall, we are unable to draw any firm conclusion because using nearest words to interpret the learned prompts could be inaccurate.

6. Conclusion

In this project, we propose four candidates approaches to compare and demonstrate the performance of each approach on adapting the foundation model CLIP to land cover classification task using EuroSAT dataset.

As shown in the result, (1) CLIP's zero-shot performance is not good on EuroSAT with only 43% F1-score. (2) The upper bound adopting linear probing training on the whole dataset for fully representation learning can achieve an upper bound of 94% F1-score, which is 51% improvement compared to zero-shot. It proves that with sufficient labelled data, even with the most simplest one linear layer added to the architecture, the CLIP model can achieve impressive performance. (3) The 16 shots linear probing can improve the performance by 38% to achieve 81% F1-score and close to the upper bound. This proves that CLIP has already learnt the generalized presentation of image in various scenarios so that it could perform well with only few shots. (4) By 16 shots prompt learning, the improvement can is similar to the 16 shots linear probing. This proves prompt learning is an auxiliary approach to deploy CLIP to downstream task while with the benefits that the output remains flexible and unlimited, which can be easily apply to new task, and gaining better transparency and explainability. (5) However, we should note the the class-specific context has trivial difference ($< 1\%$) compared to unified context, which demonstrate that for this dataset an unified context is enough, probably with the indication that different classes share the common description, and there is no need to learn extra context. (6) CoCoOp approach to focus on instance instead of class can improve the performance as well by 22% to achieve a 65% F1-score, and note that the test set contains the data and classes that it has never seen in the training, which prove its generalization ability on new class in the same dataset.

7. Future Work

As discussed in [4], they proposed another two approaches to adapt the foundation model like CLIP. First is the LayerNorm-tuning, which only trains existing Layer Normalization parameters across all Transformer layers. Second is the Adapter modules are composed of a linear down-projection, non-linearity, and linear up-projection, and are inserted inside the Transformer layers of the text encoder after the attention block.

For LayerNorm-tuning, instead of full model fine-tuning for large-scale models, we can tune a small subset of chosen parameters when the downstream data is scarce. LayerNorm applies per-element normalization across mini-batches.

For adapter modules, instead of injecting new parameters at the input or output, a third option is to inject new parameters for the downstream task within the layers of the network itself. This idea has been popularized as an efficient transfer learning method in language.

From the result of their paper, they evaluate 5 different fine-tuning baseline methods across 12 total image classification datasets and find that just tuning Layer Normalization parameters is a surprisingly effective, parameter-efficient baseline. Therefore, it is also interesting if we could adopt this approach in our task.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 2

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transform-

ers for language understanding. In *Proceedings of the 2019 Conference of the North*, 2018. 2

[3] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification, 2019. 1

[4] Konwoo Kim, Michael Laskin, Igor Mordatch, and Deepak Pathak. How to adapt your large-scale vision-and-language model, 2022. 5

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1

[6] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and PeterJ. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv: Learning,arXiv: Learning*, 2019. 2

[7] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models, 2022. 1

[8] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 1