Enhancing Zero-Shot Image Classification: A Triad Approach with Prompt Refinement, Confidence Calibration, and Ensembling

Raghav Mehta
Akridata
Raghav.mehta@akridata.com

Rakshith Sundaraiah
Akridata
Rakshith.sundaraiah@akridata.com

Sabarish Vadarevu
Akridata
Sabarish.vadarevu@akridata.com

Vijay Karamcheti
Akridata
vijay.karamcheti@akridata.com

***Abstract*—CLIP (Contrastive Language-Image Pre-training) excels in zero-shot image classification across diverse domains, making it an ideal candidate for pre-labelling unlabelled datasets. This paper introduces three pivotal enhancements designed to elevate CLIP-based pre-labeling efficacy without the need for labeled data. First, we introduce prompt refinement using a large language model (GPT-3.5-Turbo) to generate more descriptive prompts, significantly boosting accuracy on various datasets. Second, we address overconfident predictions through confidence calibration, achieving improved results without the need for a separate labeled validation set. Lastly, we leverage the inductive biases of CLIP and DINOv2 through ensembling, demonstrating a substantial boost in zero-shot labeling accuracy. Experimental results across various datasets consistently demonstrate enhanced performance, particularly in handling ambiguous classes. This work not only addresses limitations in CLIP but also provides valuable insights for advancing multimodal models in real-world applications.**

Keywords—machine learning, computer vision, classification, zero-shot, CLIP, DINOv2, ensemble models, prompt tuning

I. INTRODUCTION

This paper introduces an augmented iteration of CLIP (Contrastive Language-Image Pre-training), a pioneering multimodal model developed by OpenAI [1]. While CLIP exhibits commendable zero-shot performance, its efficacy in pre-labelling diverse datasets often falls short. Recognising this gap, we introduce three targeted techniques to bolster CLIP's adaptability and precision: refining input text prompts, calibrating confidence scores, and leveraging the image-only DINOv2 featurizer [2]. By addressing inherent challenges in pre-labelling scenarios, we equip CLIP with a nuanced understanding that exceeds its impressive zero-shot capabilities, ensuring its proficiency in diverse classification tasks.

Each of the three proposed methods separately enhances CLIP performance, starting with the nuanced refinement of input text prompts. By carefully curating prompts to describe classes of interest, CLIP's semantic understanding of the classes is significantly improved [3]. Calibrating confidence scores tackles differences in how the model perceives samples in its latent feature space and the classes to be labeled in the present task. The original pre-training establishes fixed similarities between classes, but for a specific user task, a different perspective may be needed [1].

Furthermore, we introduce an ensemble approach by incorporating the image-only DINOv2 featurizer. The varied biases from language-image pre-training and image-only contrastive training in the two models create diversity in predictions, making ensembling beneficial [2]. Through comprehensive experimentation and analysis, we showcase the collective impact of these enhancements on the CLIP model.

Our experiments across various datasets demonstrated consistent accuracy boosts. From minor to significant improvements, these methods prove pivotal for enhancing CLIP in pre-labelling classification datasets, thus setting an encouraging tone for our continued exploration.

II. PROMPT REFINEMENT

To enhance the adaptability and accuracy of CLIP for pre-labelling diverse datasets, we introduce a novel approach to prompt refinement. This technique leverages Language Models (LLMs) to curate descriptive prompts for classes, addressing nuances in language representation and ambiguity inherent in natural language. The refined prompts contribute to an improved semantic understanding of classes within the CLIP model.

A. Descriptive Prompt Generation

We use LLMs to generate descriptive prompts for each class in the dataset. By utilising the inherent language understanding capabilities of LLMs, we aim to capture the rich semantics associated with each class [4]. This involves crafting prompts that not only succinctly describe the class but also encapsulate contextual information that aids in differentiating between similar classes.

B. Ambiguity Scoring

To further refine the prompt selection process, we utilise LLMs to assign ambiguity scores to the generated prompts. These scores are determined based on multiple factors:

is_misspelled: Identifying if a prompt contains incorrect spellings, ensuring the accuracy of the language used.

is_ambiguous: Evaluating whether a prompt is ambiguous or non-ambiguous, particularly when a class name has multiple meanings.

is_generic: Assessing the specificity of the prompt, flagging prompts that may be too generic and require additional specificity.

common_score: Assigning a float value indicating the commonality of the prompt phrase, allowing for the selection of diverse and representative prompts.

C. *Refined Prompt*

Incorporating the above-mentioned scores, the LLM produces cleaned options that serve as refined prompts. These cleaned options form a curated list providing additional context about the input class name and its superclass (type of object). The list aims to enrich the understanding of the class within the multimodal CLIP framework. Multiple options are provided to ensure a comprehensive coverage of the class semantics, fostering a nuanced understanding that transcends beyond the limitations of a single prompt.

By integrating prompt refinement through LLMs, we enhance the textual input to CLIP, enabling the model to better discern and categorise diverse classes in pre-labelling scenarios. This meticulous approach contributes to the overall efficacy of CLIP in handling a wide range of classification tasks.

Selected example outputs for the "Schooner" class in the "Caltech101" dataset:

I. Ambiguity, Genericity, and proposed cleaned options for a few Caltech101 classes

Class	**is_ambiguous**	**is_generic**	**cleaned_options**
Schooner	TRUE	FALSE	['Schooner, a type of sailing ship.', 'Schooner, a type of beer glass.']
Ceiling fan	FALSE	FALSE	['ceiling fan, a type of electrical appliance']
Snoopy	TRUE	FALSE	['Snoopy, a fictional character from Peanuts comic strip.', 'Snoopy, a beagle dog.']

II. Commonality for several classes in Oxford Flowers 102 dataset

Class - Score (Commonality)
pink primrose' - Score: 0.2 (Uncommon)
'canterbury bells' - Score: 0.3 (Uncommon)
'sweet pea' - Score: 0.4 (Moderately Common)
'tiger lily' - Score: 0.4 (Moderately Common)
'bird of paradise' - Score: 0.5 (Moderately Common)
'globe thistle' - Score: 0.3 (Uncommon)
'snapdragon' - Score: 0.4 (Moderately Common)
'king protea' - Score: 0.2 (Uncommon)
'purple coneflower' - Score: 0.4 (Moderately Common)
'red ginger' - Score: 0.3 (Uncommon)
'daffodil' - Score: 0.5 (Moderately Common)
'sunflower' - Score: 0.6 (Common)

III. CONFIDENCE CALIBRATION

In our methodology, confidence scores are assigned to each sample through the scaling of the maximum predicted probability. The scaling factor employed can accommodate different types of uncertainty estimates. Specifically, we derive this factor by computing the simple average of normalized-entropies [5] obtained from final probabilities and adjusted probabilities, the latter being obtained by subtracting class-specific confidence thresholds [6] computed using CLIP predictions as pseudo-labels.

Consequently, probabilities associated with ambiguous instances are penalised more, causing them to shift towards the lower end of the confidence range. This approach facilitates the user in establishing appropriate confidence thresholds, enabling swift identification and early filtering out of ambiguous samples. Mathematically, we express confidence score (confidence) as the product of the maximum probability and the scaling factor:

confidence = max-proba × scaling_factor

where, the scaling_factor is computed as the average of the entropies from final probabilities and adjusted probabilities:

scaling_factor = average(entropy(final_probabilities), entropy(adjusted_probabilities))

IV. DINOV2 ENSEMBLE

To enhance accuracy, we train a Support Vector Classification (SVC) model using features from the DINOv2 model and pseudo-labels obtained from CLIP. The training dataset is refined by utilising confidence scores assigned to each sample, allowing us to select the top-x% of confident samples for training. Specifically, we choose the top 60% of samples from each class for model training.

The final probabilities produced by the trained model are averaged with the softmax probabilities from CLIP. This amalgamation mitigates overconfident mispredictions that may arise when the model is trained with pseudo-labels from CLIP.

When the training data adequately represents good samples for each class, prediction accuracy significantly improves. Conversely, if the training data lacks adequate good class representation, averaging the probabilities with those obtained from CLIP addresses these situations, preventing inflated probabilities for mispredictions.

final_probabilities = average(CLIP probabilities + SVC_DinoV2 probabilities)

Finally, the confidence scores for each sample are reassigned based on the obtained final probabilities. Our proposed method exhibits improved accuracy compared to predictions from CLIP alone, and the associated confidence scores aid in the improved identification and filtration of ambiguous samples.

III. Effect of prompt refinement: Classification accuracy using different prompt templates.

Dataset	**Baseline (An image of a {class_name})**	**Ours (An image of a {cleaned_option})**
Caltech101	0.826	**0.866**
Caltech256	0.814	**0.828**
Food101	0.777	**0.804**

Flowers102	0.618	**0.677**

V. Results

To demonstrate the effectiveness of our approach, we conducted experiments on several publicly available datasets and show that our approach improves or retains accuracy compared to CLIP. In all experiments, the ViT-B/32 variant of CLIP was used. When ensembling with DINOv2, the S/14 variant of the pre-trained DINOv2 featurizer is used.

IV. Effect of confidence calibration and DINOv2 ensembling, compared to zero-shot CLIP only, using the same prompts

Dataset	**Zero-shot CLIP**	**CLIP + DINOv2/SVC + Confidence calibration**
OxfordIIITPets	0.85	**0.90**
Caltech101	0.83	**0.86**
CIFAR10	0.89	**0.95**
Flowers102	0.62	**0.67**
EuroSAT	0.32	**0.45**

Table III shows the effect of prompt refinement., by comparing accuracy scores for several datasets using a baseline text prompt, using the template “An image of a {class_name}”, against the custom prompt generated through our refinement, “An image of a {cleaned_option}”. See table I for examples for the ‘cleaned_option’ generated for a few different classes. The accuracy shows a consistent and significant improvement through the prompt refinement. Table IV shows the accuracy for a subset of the dataset containing only the ambiguous classes. These ambiguous classes are identified during the prompt refinement by an LLM.

Table IV shows the effect of ensembling (DINOv2 featurizer + SVC) and confidence calibration. The accuracy improves consistently over the zero-shot CLIP baseline. In both cases, the prompts used are of the template “An image of a {class_name}”.

VI. Conclusion

Our study introduces essential enhancements to the CLIP model, addressing limitations in pre-labelling diverse datasets. Through prompt refinement, confidence calibration, and a DINOv2 ensemble approach, we augment CLIP's adaptability and precision. Experimental results demonstrate consistent accuracy improvements, especially in handling ambiguous classes. The combined impact of these techniques signifies a promising step towards achieving robust multimodal classification. As we continue to refine and explore these methodologies, our work contributes valuable insights to the ongoing advancement of multimodal models, enhancing their utility in diverse real-world applications.

References

1. Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry et al. "Learning transferable visual models from natural language supervision." In International conference on machine learning, pp. 8748-8763. PMLR, 2021.
2. Oquab, Maxime, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez et al. "Dinov2: Learning robust visual features without supervision." arXiv preprint arXiv:2304.07193 (2023).
3. Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.
4. Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
5. DeVries, Terrance, and Graham W. Taylor. "Learning confidence for out-of-distribution detection in neural networks." arXiv preprint arXiv:1802.04865 (2018).
6. Northcutt, Curtis, Lu Jiang, and Isaac Chuang. "Confident learning: Estimating uncertainty in dataset labels." Journal of Artificial Intelligence Research 70 (2021): 1373-1411.