INF473V Challenge - Image classification in semi-supervised settings

Charrin Octave
Ecole Polytechnique
octave.charrin@polytechnique.edu

Yaici Ines
Ecole Polytechnique
ines.yaici@polytechnique.edu

Abstract

Deep neural networks have gained popularity in various fields due to their reliability in image classification tasks. However, training these networks typically demands a substantial amount of labeled data. Consequently, significant preprocessing of datasets is required to ensure a properly labeled dataset for supervised learning. An alternative approach to mitigate the costly dataset preparation effort is to label only a small portion of the dataset and train the model using both labeled and unlabeled images, which are often abundant and readily available. This learning paradigm, known as semi-supervised learning, will be investigated as part of the Kaglle competition ”INF473V 2023 Challenge” [2] [3].

1. Introduction

In this report, we explore different approaches to address the challenges of limited labeled data in classification problems. We focus on three techniques: semi-supervised learning, unsupervised clustering and ensemble learning.

Through empirical evaluation and comparative analysis, we investigate the effectiveness of these approaches in improving classification performances. By exploiting their strengths and understanding their limitations, we aim to overcome the challenges posed by limited labeled data and achieve better results in classification tasks.

2. Problem Statement

The problem we are tackling is to classify a dataset that has very few labeled examples. We have 48 different classes, but only 15 labeled images for each class. However, we have a large pool of 300,000 unlabeled images that we can use for training. Our main goal is to develop a classification model that can make accurate predictions despite the limited amount of labeled data.

To achieve this, we have the following specific objectives:

– Explore semi-supervised learning techniques: Since we don’t have many labeled examples, we want to use the unlabeled data as well. We will investigate different techniques that allow us to train the model using both labeled and unlabeled data.

– Use unsupervised clustering: Clustering algorithms can help us understand the structure of the dataset and group similar images together. By incorporating these algorithms, we hope to improve the accuracy of the classification by taking into account the relationships between images within each group.

– Implement ensemble learning: Ensemble learning involves combining several models to make better predictions. This should help us improve the overall accuracy and reliability of our models.

3. Experimental approach

In our experimental approach, we first began with the analysis of our data and subsequently proceeded to test the various methods outlined earlier in order to identify the most suitable one that addresses our specific problem.

3.1. Data Description

In this section, we describe our experimental approach toward the classification problem. We will describe how we explored numerous classification techniques and how we gradually adapted them in order to constantly improve our classification accuracy.

During the analysis of our dataset, we investigated the correlation between the classes in the following way : We started by extracting the relevant features of our data using the model CLIP as we did in the clustering step. However, we noticed a distinction in the features extracted when the CLIP model was not fine-tuned 8 compared to when it was trained on our dataset 9. In our visualisation, darker shades of blue represent classes with a high degree of correlation. To quantify this correlation, we used the Euclidean distance metric, which is commonly used in traditional clustering algorithms.

We can see that certain classes show strong correlations, which poses a challenge in accurately classifying those specific classes.

3.2. Creating a consistent training and validation dataset

It was important for our investigation to ensure a consistent comparison method among the different models we were going to explore. This involved ensuring that all the models were trained on the same training dataset and validated on the same validation dataset. We accomplished this by randomly partitioning the labeled dataset using a fixed seed for the random generator. By doing so, we consistently extracted 80% of the labeled dataset as our training dataset, while the remaining 20% served as the validation dataset. This allowed us to compare the performances of our different models on the same validation dataset.

3.3. Finetuning different models

Our first approach was to finetune different models on the small amount of labeled data we had. The challenge was to find the right model with the right hyper-parameters. In Table 1, we compare different training settings that we tested. Both training dataset and validation dataset were described in the previous subsection.

We initially trained two models using a frozen backbone, either Resnet50 or VGGNet, along with a fully-connected layer that generated the output classification vector. Subsequently, we evaluated the performance of a model called CLIP from OpenAI [6], using different architectures such as ViT B/32 and ViT B/16. As shown in Table 1, the CLIP-ViT B/16 model demonstrated the best performance, without employing data augmentation, with a learning rate of 1e-7 and a weight decay of 0.01. Therefore, we decided to explore various training methods to further enhance the performance of this model.

The fundamental principle behind CLIP models is their ability to 'clip' text and images. By providing the model with a list of class names and the corresponding images, we can classify the images into 48 different classes. Consequently, the model produces a vector of size 48, which can be interpreted as the probabilities of the image belonging to each class mentioned in the input class name list.

3.4. Pseudo labeling

The first training paradigm using unlabeled data we explored was pseudo-labeling. First proposed by Lee in 2013 [5], pseudo-labeling is straight forward and follows the following 4 steps to train a model using both labeled and unlabeled data:

- Train the model on labeled data.
- Use the model to predict labels on unlabeled data.
- Combine the predicted labels with the labeled dataset.
- Train on the combination of dataset and repeat.

However this approach was not very conclusive with our CLIP-ViT B/16 model, since it was already very good at 'clipping' class names to images. In fact, the classes with already high accuracy were growing exponentially in terms of available labeled (or pseudo-labeled) data, whereas classes that were difficult to classify were not growing. This was leading to an imbalance among the different classes of the training dataset, which was probably the reason why this method was not improving the performances of our model, even if it was really effective when it came to training simpler models such as the finetuning a Resnet50 with frozen weights and a fully connected classification layer.

3.5. Using CLIP specificity

As mentioned earlier, CLIP employs a unique mechanism of 'clipping' words and images together. Consequently, the choice of words used to prompt CLIP for image classification significantly impacts the model's response. To investigate this, we conducted fine-tuning experiments on the same CLIP model using two different sets of words, and then analyzed the resulting confusion matrices.

The first set of words consisted of the class names found in the image folders themselves. The second set of words was manually crafted. We initially examined the confusion matrix of the first model (Figure 1) to identify class names that required more precision, aiming to help CLIP avoid misclassifying images in semantically similar classes. This led us to create the second model, which exhibited slightly improved performance on specific classes that were misclassified in the first model (refer to the first rows of the confusion matrix in Figure 2).

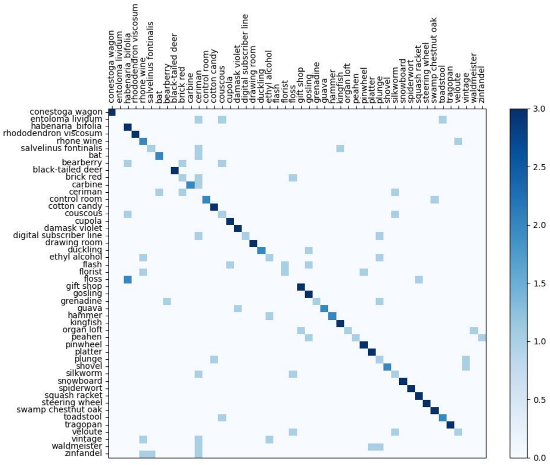

Figure 1: Confusion matrix of CLIP trained on class name list.

	data augmentation	optimizer	lr	wd	custom words	accuracy
Resnet50	✗	Adam	1e-3	0	/	0.284
Resnet50	✗	SGD	1e-3	0	/	0.519
VGGNet	✗	Adam	1e-3	0	/	0.312
VGGNet	✗	SGD	1e-3	0	/	0.277
CLIP-ViT B/32	✗	AdamW	1e-5	0	✗	0.518
CLIP-ViT B/32	✓	AdamW	1e-5	0	✗	0.521
CLIP-ViT B/32	✗	AdamW	1e-5	0	✓	0.537
CLIP-ViT B/16	✓	AdamW	1e-5	0	✗	0.537
CLIP-ViT B/16	✗	AdamW	1e-5	0.001	✗	0.551
CLIP-ViT B/16	✗	AdamW	1e-7	0.01	✗	0.568
CLIP-ViT B/16	✗	AdamW	1e-7	0.1	✗	0.542
CLIP-ViT B/16	✗	AdamW	1e-7	0.01	✓	0.583

Table 1: Comparison of different training settings on various models. (lr: learning-rate, wd: weight-decay)

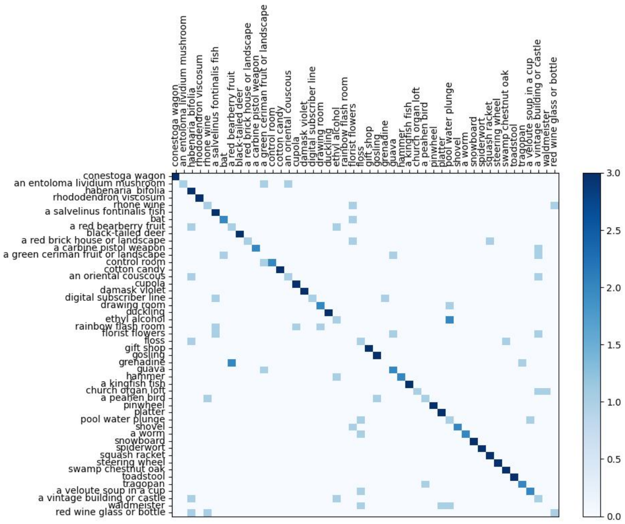

Figure 2: Confusion matrix of CLIP trained on a custom word list.

3.6. Wise finetuning

As the dataset contained images that were very different from one another even within a same class, we thought that it could be interesting to implement wise-finetuning. First introduced by Wortsman *et al.* in 2021 [7], wise finetuning consists in mixing the weights of the finetuned model with the weights of the zero-shot model such that the new weights θ are given by:

$$\theta = (1-\alpha)\theta_0 + \alpha\theta_1$$

where θ_0 are the weights of the zero-shot model and θ_1 the one of the finetuned model. By finding the right α, we are able to maintain the accuracy of the finetuned model, and to additionnaly take advantage of the precision of the zero-shot model on more general data. We found that this method helped increasing the accuracy on the validation dataset as the Figure 3 shows. This prevents the model from overfitting, which is very interesting in our case since we only have a small amount of labeled data.

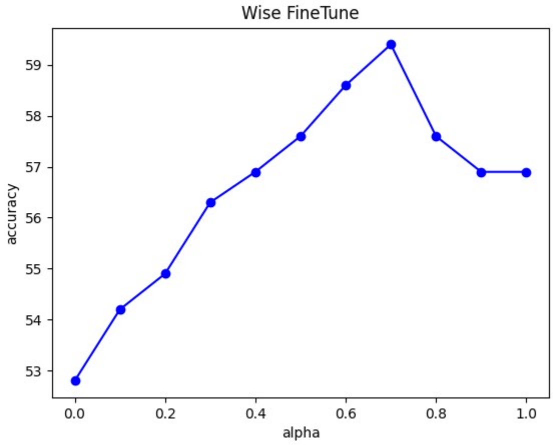

Figure 3: Accuracy with respect to α in the wise-finetuning of our CLIP-ViT B/16 trained without custom word list, AdamW optimizer, lr=1e-7 and wd=0.01.

3.7. Unsupervised clustering

The methodology uses unlabeled and labeled images to address the clustering problem. The unlabeled data analysis involves processing unlabeled images and extracting their visual features using the CLIP model. The K-means clustering algorithm is applied to group similar images based on their features, revealing patterns within the unlabeled data.

Labeled data analysis involves processing the labeled images and extracting their features using the same CLIP model. The trained K-means model assigns clusters to the labeled images, allowing the identification of the most prominent cluster within each labeled class.

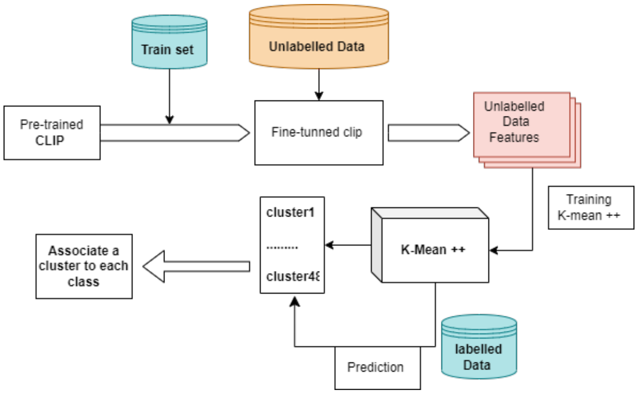

Figure 4: Ensemble learning mechanism

The general approach of the code to associate each class to a cluster using the CLIP model and the clustering algorithm is as follows:

- Fine-tune a CLIP model using the training dataset.
- Extract features from the unlabeled data using the trained CLIP model.
- Train a clustering algorithm (K-mean++) on these features.
- Extract features from the labeled data using the CLIP model.
- Predict the cluster for each class by using the trained CLIP model.
- Associate each class with the corresponding cluster by performing an argmax operation on the predictions.

By combining the results of the unlabeled and labeled data analyses, the methodology aims at solving the clustering problem by identifying representative clusters for each class.

One limitation of this methodology is the subjectivity in interpreting the identified clusters. Another input may be necessary to determine the meaningfulness and relevance of the visual patterns. Without proper guidance, the clusters may not accurately align with intended class boundaries.

Additionally, The approach of training CLIP and using K-means clustering has certain drawbacks. The training process can be time-consuming when using the entire unlabeled dataset, taking up to 3 hours. Additionally, the accuracy of the approach varies depending on the distinctiveness of the class features. It performs well for classes with distinct features but may give inconclusive results for classes with high correlation.

To improve the approach, optimization of the training process can be explored, such as using representative subsets of the unlabeled dataset or implementing dimensionality reduction techniques. Alternative clustering algorithms and the inclusion of additional features can also enhance accuracy and performance.

3.8. Ensemble learning

Ensemble learning is a method of combining the keys characteristics of multiple models to achieve better predictions. Some models may perform better on some distributions within the dataset. In our case we used ”The Bagging ensemble technique”, it generates random subsets from the dataset (some data points may appear in multiple subsets). The process starts by selecting different models that are trained independently on these subsets. Theses models are then trained together. The ensemble learning optimises the weights assigned to each model's output based on its performance and contribution.

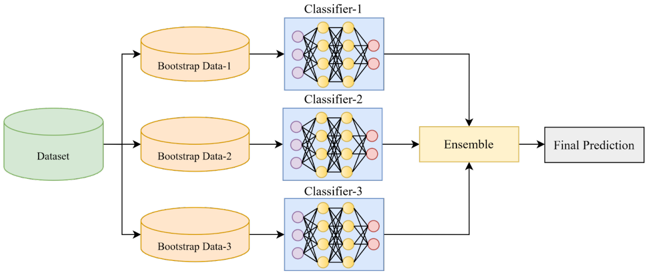

Figure 5: Ensemble learning mechanism [4]

In the prediction phase, each model generates its own predictions. These predictions are combined using the learned weights and averaging mechanism.

This technique reduces the risk of relying on the limitations or biases of a single model. The learned weights adaptively give greater importance to more accurate models, resulting in improved overall prediction performance.

3.9. Mean-Teacher

This final subsection discusses a method that we explored but were unable to fully pursue due to time constraints and implementation challenges. However, this method showed promise in developing models that could learn distinctive features from images.

Introduced by Tarvainen *et al.* in 2018 [1], consists in training simultaneously two models, one teacher and one student (Figure 6. The teacher's weights are the exponential moving average of the student's weights, so that the teacher model is an average of consecutive student models. We introduce different noises in both the teacher and the student. This can be done using random augmentations or dropout in both models. The loss is calculated by summing a classification loss for the student model and a consistency loss between the student and the teacher. The training process

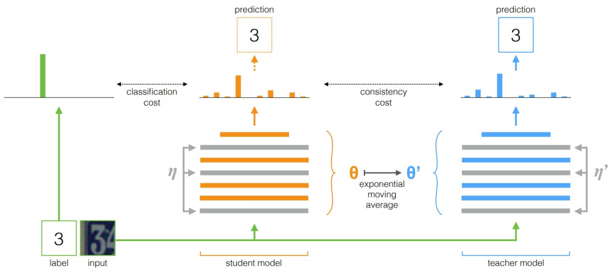

Figure 6: Mean Teacher method [1]

uses both labeled data (for classification loss) and unlabeled data (for consistency loss).

In this manner, the student model has to be increasingly precise and consistent in his predictions which helps it learn new features on images. However this training paradigm appeared to be very time consuming and not very efficient in the way we used it on our data. We trained the model for 2000 epochs during the night but only got an accuracy around 15% on the validation dataset. We would have liked to explore more deeply this method, and try for example to increase the amount of labeled data using pseudo-labeling.

4. Final results and analysis

As demonstrated in the preceding sections, the CLIP-ViT B/16 model trained without data augmentation, utilizing the AdamW optimizer with a learning rate of 1e-7, weight decay of 0.01, and employing a custom word list, achieved the highest accuracy. Moreover, the wise-finetuning approach aided in enhancing the model's accuracy while preventing overfitting on the limited training dataset, thereby maintaining performance on general images close to that of the zero-shot model.

Additionally, we observed that training CLIP with different word lists resulted in varying behaviors on specific classes. This approach offered high interpretability as we manually adjusted the precision of class denominations using custom word lists in order manipulate and specialise CLIP on precise classes.

To obtain our best-performing model for the competition, we employed an ensemble model that combined two CLIP models. Each model was trained on a different word list and further fine-tuned using the wise-finetuning method. This ensemble approach allowed us to leverage the strengths of each model, resulting in improved overall performance as we can see on Figure 7. Indeed the predictions of the ensemble model are more centered around the diagonal of the confusion matrix, which reflects the fact that the model learned how to combine the two CLIP models in order to maximize the prediction accuracy.

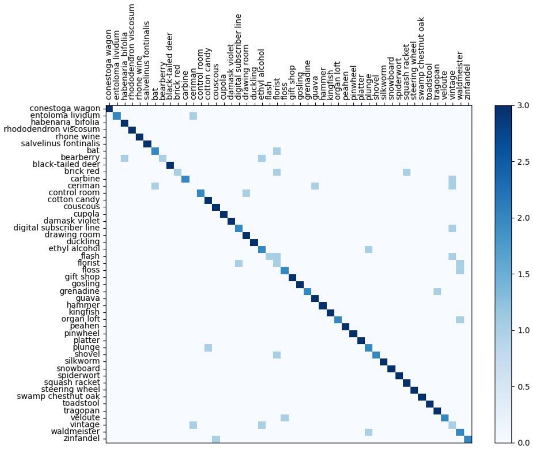

Figure 7: Confusion matrix of the final ensemble model

5. Conclusion

In conclusion, the classification problem posed several challenges that required carful considérations. Through our experimental approach, we analyzed the dataset and tested diffrent methods, taking into account the correlation between classes. We observed that certain classes exhibited strong correlation, which made accurate classification particularly challenging for those classes.

This project also amphasied valuable insights of the complexities of classification in various domains hence the importance of developing more specialized strategies to adress these specific challenges. Future research could focus on exploring advaced feature separation techniques or incorporating additional contextual information to help models like CLIP improving the classification.

For this project, Ines Yaici took responsibility for the clustering and data augmentation aspects of our investigation, while Octave Charrin handled the wise-finetuning and mean-teacher components. Both of us contributed to the initial stage of the project by finetuning various models, and we also collaborated towards the end of the project by studying and implementing ensemble learning

References

[1] Harri Valpola Antti Tarvainen. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. `https://arxiv.org/abs/1703.01780?context=stat`.

[2] Nicolas Dufour. Inf473v 2023 challenge.

[3] Nicolas Dufour. Inf473v 2023 challenge v2.

[4] Rohit Kundu. The complete guide to ensemble learning. `https://www.v7labs.com/blog/ensemble-learning`.

[5] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. 2023.

[6] OpenAI. Clip: Connecting text and images.

[7] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*, 2021. https://arxiv.org/abs/2109.01903.

6. Appendices

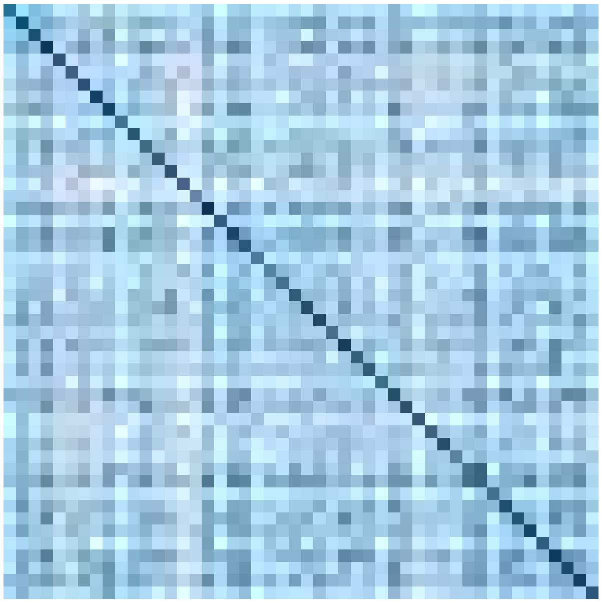

Figure 8: Correlation of the classes features extracted with a non Fine-tunned CLIP

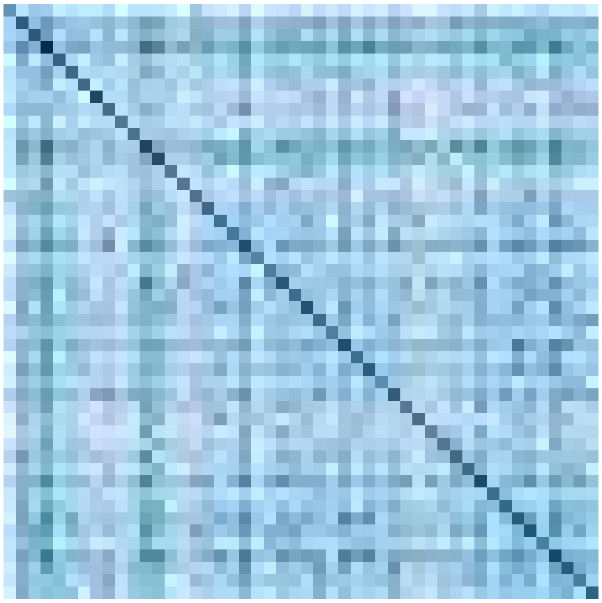

Figure 9: Correlation of the classes features with a fine-tunned CLIP