COMP 551 MiniProject 4: Reproducibility in ML

Lansu Dai
McGill University

Zichuan Guan
McGill University

Lan Xi Zhu
McGill University

Abstract

We investigated the proposed two-step unsupervised clustering method by Van Gansbeke et al. in the paper "SCAN: Learning to Classify Images without Labels". We used the provided code and pre-trained weights to verify the authors' claims on the CIFAR datasets. We successfully reproduced the claimed model performance. We studied the two-step model structure and investigated the impact of confidence threshold in fine-tuning the self-labeling step.

1 Introduction

Unsupervised image classification tasks aim to categorize unlabelled image instances into semantically meaningful clusters, e.g., where images showing similar objects would ideally be placed into the same cluster[1]. In practice, we can skip the time-consuming and costly step of labelling the training instances and allow the algorithms to recognize features from the dataset itself[1]. The goal of this project is to investigate Semantic Clustering by Adopting Nearest neighbor (SCAN), a two-step unsupervised clustering algorithm introduced in Van Gansbeke et al. 2020. The authors report optimal model performance on CIFAR10, CIFAR100, STL10, and ImageNet. Pre-trained weights at different stages of the learning routine are provided for all aforementioned datasets. We reproduced the reported test accuracy for CIFAR10 and CIFAR100 datasets through running publicly available codes associated with the paper. We ran the self-labeling step, which is the last step in the proposed model, for lower number of epochs with different hyperparameters and observe similar behaviour as reported in the paper.

2 Methodology

All content in this section cites Van Gansbeke et al. 2020.

2.1 Model descriptions

The model starts with a neural network Φ_θ with weights θ, which extract high-level features from input images through performing instance discrimination pretext task. Then for each image instance, its K-nearest neighbours are computed based on Φ_θ. It was empirically observed for all four tested datasets that an image and its assigned neighbours likely belong to the same semantic cluster. These groupings are thus used as prior to the clustering step, where the loss function is designed in favour of images being clustered with their neighbours. Another neural network Φ_η with weights η is learnt through this "Clustering Step".

An extra fine-tuning step can be added to the model following the clustering step to improve the model accuracy. This step is based on the assumption that confident predictions (with the threshold being a tunable hyperparameter) are correctly classified to the cluster they belong to, and that the noises (incorrectly assigned neighbours) tend to have less certain predictions. The neural network parameters are updated as predictions become increasingly confident. Model performances both prior to and after the self-labeling step are reported in the paper.

Pre-trained weights are provided for the pretext task step (θ), the clustering step (η), and the self-labelling step (fine-tuned η).

2.2 Datasets

Most of our experiments are performed using CIFAR10 [2], a dataset containing 60000 32x32 labelled color images. The instances are evenly distributed across 10 classes, each class having 5000 training and 1000 test instances. The CIFAR100 dataset [2] has the exact same dimension as CIFAR10, except that it has 20 super classes including a total of 100 sub classes with evenly distributed instances. Every instance therefore has two labels.

33rd Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Algorithm 1 Semantic Clustering by Adopting Nearest neighbors (SCAN)

Input: Dataset $\mathcal{D}$, Clusters $\mathcal{C}$, Task τ, Neural Nets Φ_θ and Φ_η, Neighbors $\mathcal{N}_\mathcal{D} = \{\}$.
Optimize Φ_θ with task τ. ▷ Pretext Task Step, Sec. 2.1
for $X_i \in \mathcal{D}$ **do**
    $\mathcal{N}_\mathcal{D} \leftarrow \mathcal{N}_\mathcal{D} \cup \mathcal{N}_{X_i}$, with $\mathcal{N}_{X_i} = K$ neighboring samples of $\Phi_\theta(X_i)$.
end for
while SCAN-loss decreases **do** ▷ Clustering Step, Sec. 2.2
    Update Φ_η with SCAN-loss, i.e. $\Lambda(\Phi_\eta(\mathcal{D}), \mathcal{N}_\mathcal{D}, C)$ in Eq. 2
end while
while $Len(Y)$ increases **do** ▷ Self-Labeling Step, Sec. 2.3
    Y $\leftarrow$ $(\Phi_\eta(\mathcal{D}) >$ threshold)
    Update Φ_η with cross-entropy loss, i.e. $H(\Phi_\eta(\mathcal{D}), Y)$
end while
Return: $\Phi_\eta(\mathcal{D})$ ▷ $\mathcal{D}$ is divided over C clusters

Figure 1: Pseudocode describing the structure of the SCAN model [1].

2.3 Hyperparameters

The pretext task step uses ResNet-18 backbone and the number of nearest neighbours is fixed to 20 throughout. In this project we directly use pre-trained weight for this neural network (Φ_θ) and we do not explore any hyperparameters at this step. The reported performance metrics (Table 1) after the clustering step ("SCAN-loss") are obtained with 100 training epochs and a batch size of 128. The self-labelling step requires another 200 training epochs with a batch size of 1000, according to the provided codes. The confidence threshold (see Section 2.1) was set to 0.99. It has been shown in the paper that the model performance does not vary significantly with a threshold between 0.9 and 0.99. Due to the limitation of computational resources in this project, aside from running the model completely with provided pre-trained weights, we run the last training step with lower epochs to investigate the influence on model performance. We also briefly study the effect of varying confidence threshold in the self-labelling step using pre-trained weights for SCAN-loss.

Dataset	Step	ACC
CIFAR10	SCAN-loss	81.6
	Self-labeling	88.3
CIFAR100	SCAN-loss	44.0
	Self-labeling	50.7
STL10	SCAN-loss	79.2
	Self-labeling	80.9

Table 1: A subset of model performances reported by Van Gansbeke et al. 2020.

3 Results

3.1 Results reproducing original paper

Using provided pre-trained weights at all three steps, we were able to obtain the exact same performance metrics as reported in the paper for CIFAR10 and CIFAR100 (also called CIFAR20 in the implementation).

3.2 Self-labeling step hyperparameters

Starting from an accuracy of 81.6% on the CIFAR10 dataset following the pre-trained clustering step (Table 1), we investigated how a self-labelling step with only 50 epochs (vs. 200 epochs in Van Gansbeke et al. 2020) improves the model performance upon the SCAN-loss clustering step. Keeping all other hyperparameters unchanged, we obtained a test accuracy of 86.65% (comparing to the reported accuracy of 88.3% using 200 epochs). This agrees with the results presented in the paper, in particular Figure 2, where the accuracy rapidly increases at early stage of the training process.

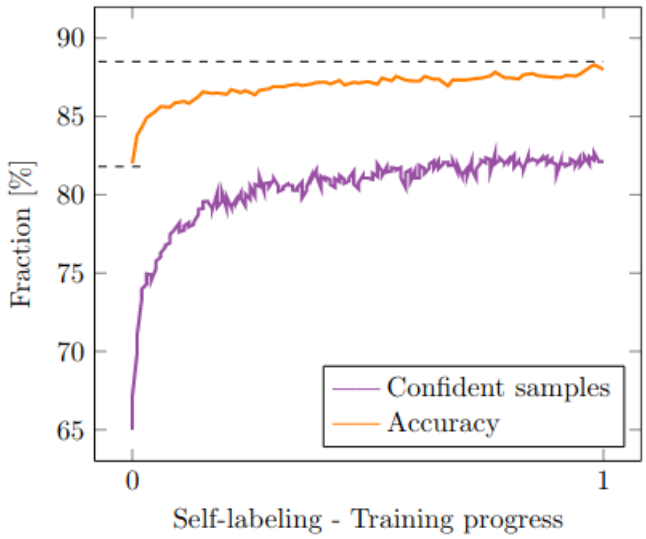

Figure 2: Variation in number of confident ($p \geq 0.99$) samples and test accuracy as a function of self-labeling training progress (200 epochs maximum) for the CIFAR10 dataset. (Van Gansbeke et al. 2020, Figure 4)

In the self-labeling step, the confident predictions in each cluster are referred to as “prototypes". We observe that the prototypes output after 50 epochs (Figure 3, first row) do not match with the ground-truth classes. In particular, there are two images of horses and 0 image of deer. Surprisingly, the pre-trained model with an accuracy of 88.3% (consistent with the value reported in Table 1) equally fails to output a prototype of deer (Figure 3, last row).

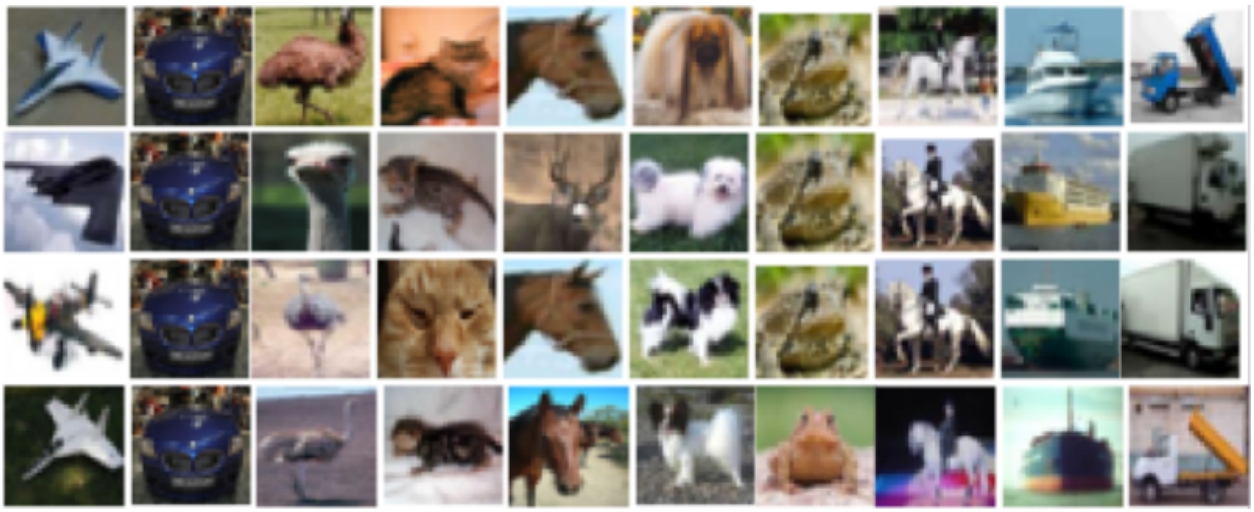

Figure 3: Prototypes output by the self-labeling step performed on CIFAR10 dataset. First row: 50 epochs, $p \geq 0.90$; second row: 50 epochs, $p \geq 0.95$; third row: 50 epochs, $p \geq 0.99$; fourth row: 200 epochs, $p \geq 0.99$ (pre-trained weight).

We have also trained the self-labeling step using a confidence threshold of 0.9 and 0.95 with 50 epochs, achieving test accuracy of 86.83% and 87.25%, respectively. The model performance stays relatively constant with varying confidence threshold, which agrees with what the paper claims for 200-epoch training (Section 2.3). In terms of prototypes, varying confidence threshold results in partially different sets of images (Figure 3, second and third rows).

4 Discussion

4.1 Prototypes

It is rather unexpected to observe, at multiple runs including the pre-trained optimal model, a set of prototypes that do not align with ground-truth classes. As mentioned earlier in Section 2.1, the self-labeling step assumes that confident predictions tend to be correct [1]. Our experimental observation suggests that exceptions are likely not rare for visually similar objects. It is also worth noting that the one-to-one correspondence from one row to another in Figure 3 does not necessarily propagates to the clusters they represent (i.e., the image of a horse does not necessarily represent the "horse" cluster).

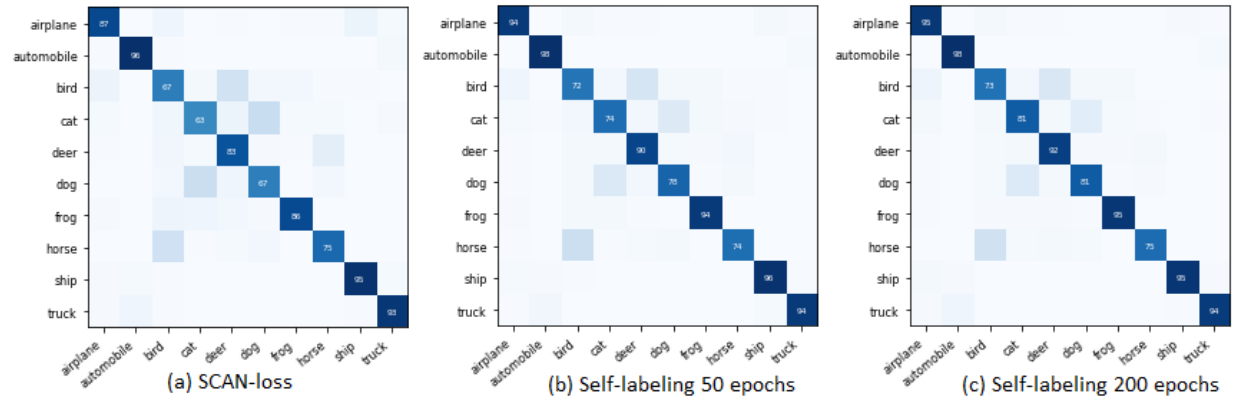

Figure 4: Confusion matrices on test set obtained with a) SCAN-loss only (ACC=81.6%), b) 50-epoch self-labeling (ACC=86.65%), and c) 200-epoch self-labeling (ACC=88.3%).

Comparing panel c) with panel a) in Figure 4, the improvement in model performance with applying the self-labeling step is expressed through paler off-diagonal entries. Amongst the four off-diagonal entries with the deepest shade, we notice that birds (all "bird" class prototypes are ostriches-like birds) are easily confused with horses, while deers tend to be confused with birds. This suggests that the absence of "deer" class in the prototypes might be the consequence of multiple confusions.

4.2 Difficulties and challenges

Since the model is divided into three steps, it was particularly difficult to explore the model parameters specific to intermediate steps with limited computational resources. For example, if we wish to investigate how decreasing the number of epochs at the clustering step influences model accuracy, we would have to run the subsequent self-labeling step with the default setting of 200 epochs for the purpose of controlling variables. Thus, we were only able to study the model parameters of the self-labeling step thanks to the pre-trained weights provided for the previous steps.

5 Conclusion

Van Gansbeke et al. (2020) introduces an unsupervised image-clustering algorithm that achieve an accuracy of 88.3% on the CIFAR10 dataset and an accuracy of 50.7% on the CIFAR100 dataset. The GitHub repository containing relevant codes provides detailed tutorial for running the experiments mentioned in Van Gansbeke et al. (2020) that we were able to follow. We reproduced the exact same performance metrics using the codes and pre-trained weights that the authors have made publicly available throughout. Using only pre-trained weights for the pretext task step and the clustering step, we ran the self-labeling step using modified hyperparameters with lower number of training epochs (due to computational resources limitation). After 50 epochs of training, the model achieves an accuracy of 86.65%, <2%

lower than the accuracy obtained after 200 epochs. Changing confidence threshold to 0.90/0.95 results in an accuracy of 86.83%/87.25% respectively, which is close to the accuracy of 86.65% obtained at a threshold value of 0.99 (model default). Due to various limitations, we were unable to reproduce most of the experiments and hyperparameter tuning mentioned in the paper. It would also be of interest if one could visualize the high-level features that are used to assign an image to its nearest neighbours.

Statement of contribution

Everyone contributed equally to the coding component and the project write-up. We have not assigned specific part of the project to individual group member.

References

[1] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "Scan: Learning to classify images without labels," in *European Conference on Computer Vision*, pp. 268–285, Springer, 2020.

[2] A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., 2009.