CS231n - Classifying dogs using PAWS

Umesh Thillaivasan
Department of Computer Science
Stanford University, Stanford Center for Professional Development
uthillai@stanford.edu

Abstract

*This paper explores the application of a novel method of learning by **p**redicting view **a**ssignments **w**ith support **s**amples (PAWS) on fine-grain datasets to understand how well this semi-supervised method using spares labels works for classification tasks when classes share many similar features due to being in the same category (e.g., one dog species from another).*

The method trains a model to minimize a consistency loss, which ensures that different views of the same unlabeled instance are assigned similar pseudo-labels. The pseudo-labels are generated non-parametrically, by comparing the representations of the image views to those of a set of randomly sampled labeled images. The distance between the view representations and labeled representations is used to provide a weighting over class labels, which is interpreted as a soft pseudo-label. By non-parametrically incorporating labeled samples in this way, PAWS extends the distance-metric loss used in self-supervised methods such as BYOL and SwAV to the semi-supervised setting.

Furthermore, this paper focuses on the performance of PAWS top-1 accuracy when fine-tuning fine-grain datasets on varying ratios of labeled to unlabeled images after pre-training on ImageNet. For fine-grain classification, PAWS went from 59.42% top-1 when using 1:1 labeled to unlabeled fine-tune training datasets, to 52.18% top-1 when using a 1:25 ratio, showing that PAWS should be explored further for fine-grain classification problems with sparsely labeled datasets.

1. Introduction

When training models for certain applications such as computer vision classification, large amounts of real-world data with labels is often needed to train classifiers so that features in the data can be learned to accurately predict classes for new images. These are commonly supervised learning methods where the training data has labels of the ground-truth. The CIFAR-10 [6] and ImageNet Large Scale Visual Recognition Challenge (ImageNet) [10] provided massive labeled datasets to help reduce the burden of expensively collecting and labeling data, and has allowed researchers to compare progress in detection across a wider variety of objects as well as use for pre-training.

However, not all classification applications have a readily available datasets. Instead, these datasets need to be created and are commonly sparsely labeled meaning that the ratio of labeled images to unlabeled images can be significant. Having sparse labels can mean that it's harder for your classifier to accurately predict each class, and this problem is further exacerbated when performing fine-grain classification using sparsely labeled datasets where differences in classes can be much more subtle such as the difference between classifying a dog from a car versus a dog breed from another dog breed.

How to learn with less labeled data is an important computer vision and machine learning research area. One popular approach for learning with few labels is to first perform unsupervised pre-training on a large dataset followed by supervised fine-tuning on the small set of available labels. Self-supervised methods follow similar training methodologies, but also requires substantially more computational effort than other approaches and does not make use of labeled data when it is available. If labeled data is available, alternative methods have used them to generate pseudo-labels for the unlabeled data, and then train a model using the labeled and pseudo-labeled data.

The method that PAWS implements to train a model is to minimize the consistency loss to ensure that views of the same unlabeled image are assigned the same pseudo-labels. The pseudo-labels are generated by comparing the representations of the image views to those of a set of randomly sampled labeled images. The distance between the view representations and labeled representations is used to provide a weighting over class labels, which we interpret as a soft pseudo-label. PAWS extends the distance-metric loss in self-supervised methods such as **B**ootstrap **Y**our **O**wn **L**atent (BYOL) [5] and **Sw**apping **A**ssignments between multiple **V**iews of the same image (SwAV) [2] to the semi-

supervised setting.

Before exploring how PAWS performs on new datasets where classes are within the same category, I started with the baseline implementation of PAWS trained for 600 epochs with a single-GPU on the CIFAR-10 dataset [6], then evaluate the nearest-neighbours performance of the model on a single GPU to better familiarize myself with the work of past publications, research, and methodology. After testing the initial code on the CIFAR-10 dataset to replicate results and understanding, I look to use the PAWS weights pre-trained on the full ImageNet 1000-class dataset and fine-tune on the fine-grain 120-class ImageNet dogs dataset to perform a study on how accuracy changes based on the ratio of labeled to unlabeled training data on fine-grain datasets.

This application-based project is aimed at exploring how a novel method of learning by **P**redicting view **A**ssignments **W**ith support **S**amples (PAWS) [1] performs on fine-grain datasets using sparsely labeled data to understand how well this methodology works for classification tasks when classes share many similar features due to being in the same category, (e.g., one dog breed from another dog breed).

The input to our algorithm are images. We then use an ResNet-50 encoder network and then a Soft Nearest Neighbours similarity classifier to output a predicted class with saliency maps [4] and confusion matrices [8].

Our results show that after pre-training on ImageNet and fine-tuning using the fine-grain dataset with as big as 1:25 labeled to unlabeled data ratio, PAWS can still be used as an effective classifier. These findings suggest that PAWS can use sparsely labeled data to effectively learn a new class, especially when that class can have minor differences to other classes as they all belong to the same class category. Applications include automation and manufacturing where new datasets need to be created for defect detection in production, but generating and labeling large datasets is costly, as well as datasets are much more likely to be fine-grain sharing similar features.

2. Related Work

As discussed earlier, the motivation for this paper is to determine how to accurately classify very similar objects in the same category when data is available, but labels are sparse.

In a literature review, there are a few approached to classify objects with few, some, or no labeled data: semi-supervised learning, few-shot learning, self-supervised learning.

PAWS extends the distance-metric loss in self-supervised methods such as **B**ootstrap **Y**our **O**wn **L**atent (BYOL) [5] and **Sw**apping **A**ssignments between multiple **V**iews of the same image (SwAV) [2] to the semi-supervised setting.

BYOL is a self-supervised learning approach using two networks to learn: the online and target network. Given an image, the online and target networks produce a prediction, and then objective of BYOL is to minimize the similarity loss between the two network predictions [5]. This minimization of the similarity loss is a key basis for PAWS.

SwAV is another key distance-metric loss that focuses on minimizing distance between different augmentations or views of the same image and assigning a cluster or class of one view based on the compared view [2]. This multi-crop of the input image is a key component in the PAWS methodology.

SimCLR is another framework for contrastive learning representations by maximizing agreement between different augmentations of the same input image. This is a key component of the PAWS approach which also leverages the contrastive loss [3]

An additional idea important for understanding PAWS is semi-supervised methods related to generating pseudo-labels for unlabeled samples. Pseudo-Label [7], is a method that first trains a model on the labeled data and then uses this model to assign pseudo-labels to unlabeled data, and then retrain the entire model again on the labeled and unlabeled data together. There are several similar methods of generating pseudo labels for the unlabeled data, and then train a model using the labeled and pseudo-labeled data [14], [9].

Another important component of PAWS is the Student-Teacher network. The approach is to use the teacher to assign pseudo-labels to unlabeled data which are used to train the student part of the network. The goal is for the student to minimize it's prediction error to the teacher's pseudo-label [13], [11].

For two reasons, semi-supervised PAWS approach was pursued: (1) training and testing using PAWS claims to be highly computationally efficient in comparison to other methods while still maintaining high performance, and (2) it is realistic and a desirable pipeline in the real-world setting to collect 2D images of objects and label a small amount of data to classify more data.

3. Dataset and Features

We currently work with 3 datasets: the CIFAR-10, the full ImageNet, and ImageNet fine-grain (Task 3) datasets. Examples of images from each of these 3 datasets can be seen in 1. The original PAWS paper performs experiments and pre-training benchmarks using ImageNet and has provided baseline weights to use for experimentation and validation.

The CIFAR-10 dataset consists of 60,000, 32x32 colour images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images [6]. For our baseline on CIFAR-10, we follow the same PAWS implementation of using 4000 labeled training images

Figure 1. Example of 2 classes and images from each of the 3 datasets used in this work. Images have different resolutions and aspect ratios, therefor appear blurry in this image. For CIFAR-10 dataset, first class is *car*, second class is *ship*. For ImageNet dataset, first image is *soccer ball*, second class is *piano*. For the fine-grain dataset, first class is *Bernese mountain dog*, the second class is *Papillon dog*.

Similarly, to baseline PAWS with existing ImageNet implementations, we use the full ImageNet dataset which has over 1.2M images 500x370 colour images and 1,000 classes [10] of varying resolution. For pre-training we use the 1% labeled data weight provided by the PAWS paper. The ImageNet fine-grain subset has 20,580 images and 120 classes of dog species. We split this into 90% training and 10% validation. The training images are split into two halves where one half is always used for unlabelled data, and the other half in incrementally added based on the number of labeled images based on the experiment. This large number of categories should make it an interesting dataset for subordinate categorization and also allows the unlabeled dataset to remain consistent for all experiments.

As a part of the training methodology, data is augmented using color jitter, random cropping, and random horizontal flipping as applied in the SimCLR approach [3] to ensure that initial baselines can be replicated.

4. Methodology

The PAWS methodology [1] starts with a large dataset of unlabeled data D, and a small support dataset of annotated images S, where the number of labeled images is much smaller than unlabeled images. During pre-training, the small amount of labeled data S is used with the unlabeled data D to learn image representations. After pre-training, the model is fine-tuned with only labeled data S.

A two-view pre-training approach is shown in Figure 2 where, using random data augmentations such as color jitter, random horizontal flipping, random cropping, two views (or multiple crops) of each unlabeled image are generated: a positive view and an anchor view. We train this network by assigning a soft pseudo-label to both the anchor and positive views and then minimizing the cross-entropy loss between the two views.

First, pseudo-labels are assigned by using a Soft Nearest Neighbours similarity classifier given by:

$$\pi_d(z_i, z_S) = \sum_{z_{sj}, y_j} \left(\frac{d(z_i, z_{sj})}{\sum d(z_i, z_{sk})}\right) y_i \tag{1}$$

where y_j is the one-hot ground truth label vector, z_i is the i^{th} representation of mini-batch z.

Then, for similarity metrics and predictions, PAWS uses the exponential temperature-scaled cosine so the prediction for the anchor and the positive view can be calculated using the softmax with temperature τ:

$$p_i = \pi_d(z_i, z_s) = \sigma_\tau(z_i z_S^T) y_S \tag{2}$$

where $\sigma_\tau()$ is the softmax with temperature $\tau > 0$ and p_i is the prediction for representation z_i and p_i^+ is the positive prediction representation for z_i^+.

PAWS also addresses a representation collapse problem where all data is assigned to one class but using the Sinkhorn-Knopp normalization and sharpening the the predictions to make them more salient and more confident. PAWS defines a sharpening function ρ given by:

$$\rho(p_i)]_k = \frac{[p_i]_k^{1/T}}{\sum_{j=1}^{K} [p_i]_j^{1/T}} \tag{3}$$

The objective function to be minimized when training the encoder is:

$$\frac{1}{2n} \sum_{i=1}^{n} (H(\rho(p_i^+), p_i) + H(\rho p_i), p_i^+)) - H(\bar{p}) \tag{4}$$

where p_i is the prediction for representation z_i and p_i^+ is the positive prediction representation for z_i^+. The prediction of one view is compared with the sharpened prediction of the other view. When training the encoder, we want the p_i and p_i^+ predictions of the two or more views of the same image to be very similar.

PAWS also uses a mean entropy maximization (ME-MAX) regularization term to maximize the entropy of $H(\bar{p})$.

PAWS authors released a paper and codebase that is the basis for this research. We build upon this codebase to add and extend functionality for fine-grain classification. Specifically we introduce saliency map generation, confusion matrices generation, and modifying base scripts to extend training from just being compatible with CIFAR-10

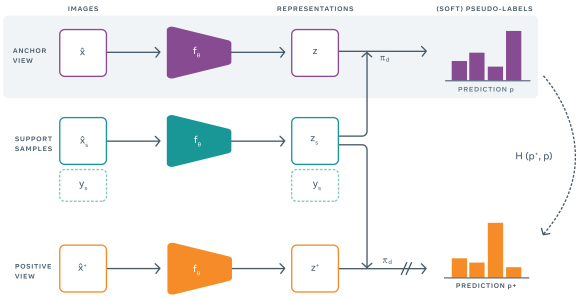

Figure 2. The PAWS method assigns soft pseudo-labels to an anchor view of an image and an associated positive view, and subsequently minimizes the cross-entropy H between them. The soft pseudo-labels are generated using a differentiable similarity classifier π_d that measures the similarity to a mini-batch of labeled support samples, and outputs a soft class distribution. Positive views are created using data-augmentations of the anchor view [1].

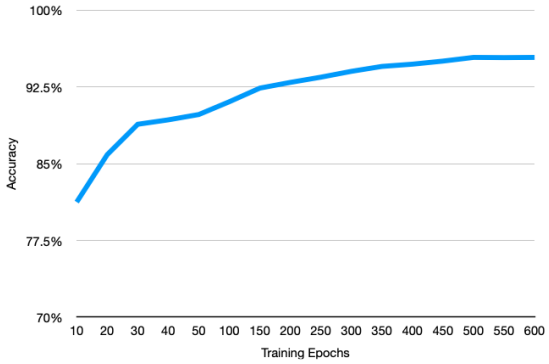

Figure 3. Baseline training on CIFAR-10 dataset using a WideResNet-28-2 architecture and pre-training on 600 epochs. We report the Top-1 class accuracy of 95.38% when training on 8% labeled data.

and ImageNet to being able to split and setup datasets as needed to run labeled to unlabeled ratios for pre-training, fine-tuning, and soft nearest neighbours classification.

5. Experiments and Results

5.1. Baselines

Before exploring how PAWS performs on new datasets where classes are within the same category, we start with the baseline implementation of PAWS trained for 600 epochs with a single-GPU on the CIFAR-10 dataset, then evaluate the nearest-neighbours performance of the model on a single GPU, with and without fine-tuning, to better familiarize ourselves with the work of past publications, research, and methodology.

Figure 3 and 4 show our baseline results on the CIFAR-10 dataset. For the baseline, we replicate the paper experiments [1]: To construct the different image views, we generating two large (global) crops (32 × 32), and six small (local) crops (18 × 18) of each unlabeled image. We use a batch-size of 256. To construct the support mini-batch at each iteration, we also randomly sample 640 images, comprising 10 classes and 64 images per class, from the labeled set, and apply default PAWS label smoothing factor of 0.1. For all sampled images (both unlabeled images and support images) we apply the basic set of SimCLR data-augmentations, specifically, random crop, horizontal flip, and color distortion. We also generate two views of each sampled support image.

We first pre-train a network using PAWS on CIFAR-10 using a 92% unsupervised, 8% supervised split with access to 4000 labels, and then report the nearest-neighbour classification accuracy on the test set using the 4000 labeled training images as support. Training was done using a Wide ResNet-28-2 encoder without dropout. It contained a 3-layer MLP projection head, consisting of three fully-connected layers of dimension 128, and Batch Normalization applied to the hidden layers.

For pre-training, we use the same PAWS implementations of a LARS optimizer with a momentum value of 0.9, weight decay 10e−6, cosine-similarity temperature of τ = 0.1 and target sharpening temperature of T = 0.25. All 10 classes were used per batch size of 256. These choices of hyperparameters were selected to ensure baselines would match as well as the original authors provided the optimized hyperparameters as a part of their work.

Initial baseline results are inline with reported PAWS performance, and we were able to get the code base successfully running with a single GPU.

The next baseline experiment was to fine-tune and evaluate PAWS on ImageNet. For reproducibility, the PAWS paper provides full checkpoints for the pre-trained models which contains the backbone, projection head, and prediction head weights. We used the 300th epoch, ResNet-50 pretrained weights on 1% labeled ImageNet data. We then fine-tuned using the provided training and validation splits, and achieved 69.792% validation accuracy which is inline with the expected paper accuracies.

5.2. Fine-grain experiments

Now that we have successfully implemented PAWS and been able to generate baselines and reproduce paper expected results, we shifted focus to the core objective of understanding how PAWS performs on the fine-grain classification dataset of 120 dog breed classes while also experimenting with the ratio of labeled to unlabeled data. We explore what the labeled to unlabeled ratio is needed to provide high accuracy on similar fine-grain classifica-

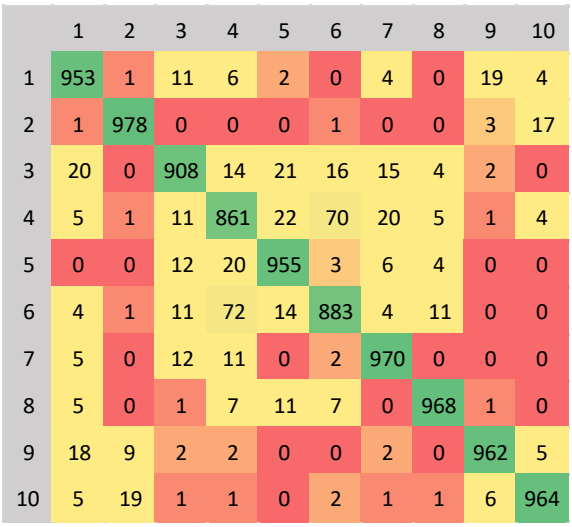

Figure 4. CIFAR10 SNN heat map confusion matrix (red is low, green is high). We report the Top-1 class accuracy of 95.38% when training on 8% labeled data.

tion datasets. Leveraging lessons from class, we also use saliency maps to visualize how the fine-tuning of the network changes the representations learned by comparing saliency maps after training and tuning the network. This will help understand how training and fine-tuning impacts what the model learns.

We perform 5 experiments where fine-tune the PAWS ResNet-50 architecture using the ImageNet fine-grain dataset on dogs. We use the ImageNet pre-trained weights trained on 1% labeled data of the ImageNet dataset. These weights were provided by the paper and what was used for one for the baselines.

We then setup several experiments where we first split the 20,580 image dataset into 90% training and 10% validation. To keep the number of unlabelled images constant for all experiments, the training dataset is then split in half. One half of 9,261 images are set aside as always unlabeled data. To create the per-experiment dataset, the ratio of labeled images is then added from the other half of the training data. For example, when performing a 1:1 ratio experiment, 9,261 unlabelled images and 9,261 labeled images were used to create the training dataset of 18,522 images. Similarly, when performing a 1:25 ratio experiment, 9,261 unlabeled and 370 labeled images were used to create the training dataset of 9,631 images.

The metrics for our experiments are both quantitative and qualitative. We use validation accuracy, SNN Top 1 and Top 5 accuracies, and confusion matrices for quantitative evaluation. We use saliency maps for qualitative analysis.

For each experiment, we fine-tune the network on the appropriate sparse dataset for 30 epochs, and record the training and validation accuracy. After fine-tuning, we then performed SNN on the fine-tuned model to return Top 1 and Top 5 results. Table 1 summarizes the 5 experiments and results of labeled to unlabeled ratios. We then generate saliency maps for each image, and a confusion matrix when performing SNN to add interpretability to the experiments and discussion.

Ratio	FT Train Acc.	FT Val Acc.	SNN Top 1	SNN Top 5
1:1	87.56%	83.30%	59.42%	87.13%
1:2	87.51%	81.89%	56.45%	85.72%
1:5	87.40%	80.96%	53.70%	83.73%
1:10	87.34%	80.86%	52.52%	83.08%
1:25	86.80%	81.20%	52.18%	81.63%

Table 1. Experimental results of varying ratio of labeled to unlabeled data from 1:1 to 1:25 ratio when fine-tuning PAWS model using the 120-class fine-grain dog ImageNet dataset for 30 epochs.

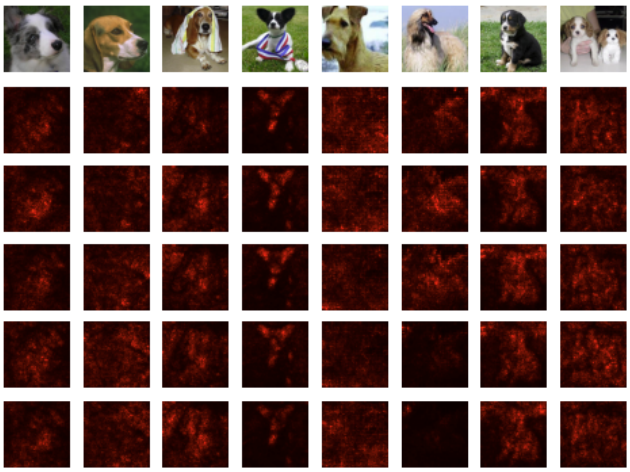

Figure 5. Saliency map of 8 classes of dogs showing raw image without data augmentations as the first row, and following rows are saliency maps of the respective image from the SNN using the model fine-tuned with varying sparse labels. First saliency row is 1:1. Last saliency row is 1:25.

6. Discussion

By leveraging a small labeled support set during pre-training, PAWS achieves competitive classification accuracy for semi-supervised problems as demonstrated in the original paper and replicated during the baseline experiments using CIFAR-10 and ImageNet.

When moving to fine-grain classification, PAWS does experience a 10% drop in SNN Top 1 accuracy as compared to the larger ImageNet dataset, but it reasonable to expect a performance reduction when classes are far more similar in features.

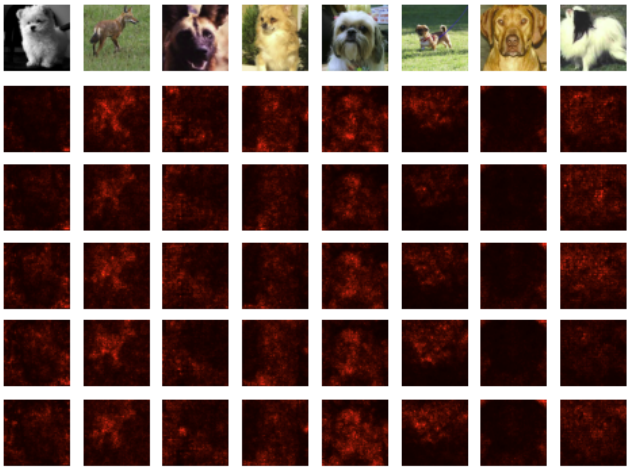

Figure 6. Saliency map of 8 classes of dogs showing raw image without data augmentations as the first row, and following rows are saliency maps of the respective image from the SNN using the model fine-tuned with varying sparse labels. First saliency row is 1:1. Last saliency row is 1:25.

The labeled to unlabeled ratio ablation experiments yielded very interesting results in that as the ratio of labeled to unlabeled images went from a 1:1 to an extreme 1:25 ratio (i.e., using sparsely labeled datasets) the fine-tune validation accuracy dropped approximately 2% while the SNN Top 1 accuracy dropped approximately 7% from 59% to 52%, which is significantly better than random chance of getting the correct class out of 120 classes (0.83%).

What's also interesting is observing how the saliency maps for the same image evolve across different ratios of support sets as seen in the Figure 5 third-last column showing a more distinguishable representation losing saliency as sparsity increases (i.e., moving down row by row). Figure 5 and Figure 6 both show examples of 16 different classes of dog in the fine-grain dataset. Each row below the raw image corresponds to the respective experimental ratio (i.e., first saliency map row corresponds to 1:1 ratio experiment).

An interesting observation is that nearly all saliency maps are most representative of the raw image with a 1:1 ratio, with the exception of the first class in Figure 6 of the *Maltese Dog*, which has a nearly non-activated saliency map. This is also true for the seventh image in Figure 6 of the *Rhodesian ridgeback* dog which had nearly no activation of the dog's face, but instead activated around the face to create an activation silhouette. It makes sense that the higher the number of labeled images, the better the network will learn the features and produce a more accurate saliency map, however, it's interesting that even with a 1:10 ratio, many raw images still have discernible saliency maps that represent the input image which supports that the SNN Top 1 accuracy did not diminish too significantly as sparsity was introduced.

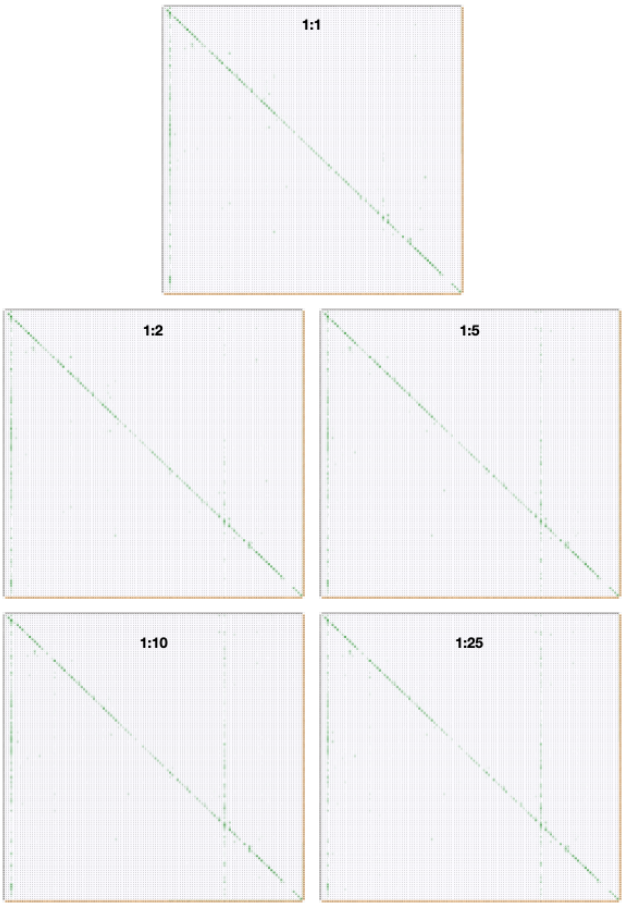

Figure 7. Summary of each experiment's SNN confusion matrix after fine-tuning with the fine-grain dogs dataset. As the sparser the dataset becomes (i.e., the ratio of labeled to unlabeled decreases) the more off-diagonal predictions are made.

Another interesting observation can be see from the SNN confusion matrices in Figure 7. In accordance to the SNN Top 1 accuracies, the respective heat map confusion matrices show most classifications are along the diagonal, but begin to fall off-diagonal as more sparsity is introduced. This is most clearly seen for class 3, *Maltese Dog* and class 89 *Bernese Mountain Dog*. We sample the training data for these classes and visualize raw images seen in Figure 8. It's not immediately clear why so many of the 120 classes falsely predict an image as either a Maltese dog (class 3) or a Bernese mountain dog (class 89), but our hypothesis is that both classes share very similar colours and general features as many other dog breeds as they are fine-grain classes. This can be further supported when observing that the saliency maps get less distinct as sparsity in introduced. It could be possible that for certain images, when performing SNN with the sparsely trained models, the network cannot mean-

Figure 8. Sample of class 3, Maltese dogs, and class 89, Bernese mountain dogs which were the classes that generated the most false predictions.

ingfully separate these less salient inputs and therefore they cluster most closely with class 3 or 89. Upon further investigation, it can be noted that each of the 120 classes in the fine-grain dataset has a minimum of 148 images, a maximum of 252 images, and the average is 171 images. Class 3 and class 89 have 252 and 218 images, respectively, which could also slightly bias the network as a more probably class prediction.

7. Conclusions and Future Work

This application-based project is aimed at exploring how a novel method of learning by **P**redicting view **A**ssignments **W**ith support **S**amples (PAWS) [1] performs on fine-grain datasets using sparsely labeled data to understand how well this methodology works for classification tasks when classes share many similar features due to being in the same category, (e.g., one dog breed from another dog breed).

Using an ResNet-50 encoder network and then a Soft Nearest Neighbours similarity classifier to output a predicted class on an image with saliency maps and confusion matrices, our results show that after pre-training on ImageNet and fine-tuning using the fine-grain dataset with as big as 1:25 labeled to unlabeled data ratio, PAWS can still be used as an effective classifier on fine-grain classification tasks.

These findings suggest that PAWS can use sparsely labeled data to effectively learn a new class, especially when that class can have minor differences to other classes as they all belong to the same class category, however performance does noticeably diminish when using very sparsely labeled datasets combined with very similar classes.

For future work, possible applications that we are interested in applying PAWS towards in the future are within the automation and manufacturing industry. New datasets often need to be created to take advantage of deep learning for computer vision such as defect detection in production, but generating and labeling large datasets is costly, as well as datasets are much more likely to be fine-grain sharing similar features. PAWS may be a viable method to add efficiencies to this process.

A final observation is that the fine-grain dogs dataset is a subset of the full ImageNet dataset. This means that the pre-training weights may have learned some representations and features from images within these 120 dog classes, improving the accuracies during fine-tuning. To support these observations, future work is needed to fine-tune PAWS on different fine-grain datasets such as CUB-200-2011 [12].

References

[1] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8443–8452, 2021. 2, 3, 4, 7

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 1, 2

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 3

[4] CS231n. Cs231n saliency maps from assignment 2, 2022. 2

[5] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 1, 2

[6] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 1, 2

[7] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. 2

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 2

[9] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568, 2021. 2

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1, 3

[11] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve

semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 2

[12] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 7

[13] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[14] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 33:3833–3845, 2020. 2