FEW-SHOT LEARNING FOR DERMATOLOGICAL DISEASE DIAGNOSIS

A Thesis
Presented to
The Academic Faculty

By

Viraj Uday Prabhu

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science in the
College of Computing

Georgia Institute of Technology

May 2019

Copyright © Viraj Uday Prabhu 2019

FEW-SHOT LEARNING FOR DERMATOLOGICAL DISEASE DIAGNOSIS

Approved by:

Dr. Devi Parikh, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Dhruv Batra
School of Interactive Computing
Georgia Institute of Technology

Dr. Stefan Lee
School of Interactive Computing
Georgia Institute of Technology

Date Approved: April 20, 2019

Traveler, there is no path.

The path is made by walking.

Antonio Machado

To my parents

ACKNOWLEDGEMENTS

I'm grateful to many people for their guidance and support through the roller coaster that is grad school. First off, I'm immensely grateful to my advisor Devi Parikh for her mentorship every step of the way. Devi's supportiveness, clarity of thought, and ability to think through complex problems, both research and otherwise, have been invaluable to me through this process, and her remarkable efficiency has become a lifelong goal to emulate. I'm also grateful to Dhruv Batra, for giving me my start in research, and for being a constant source of guidance and perspective along the way. I'm also grateful to Stefan Lee for mentoring me on my first research project, introducing me to several excellent board games, and for making our lab get-togethers lively. I also owe thanks to Anitha Kannan, for providing constant motivation and encouragement and teaching me to enjoy the process of research.

The CVMLP lab at Georgia Tech (and previously Virginia Tech) has been a great environment to work in, and I've been lucky to have had many amazing and supportive labmates. I'm grateful to Arjun, Deshraj, Prithvi, Rama, Ram, Ashwin, Nirbhay, Samyak, Aishwarya, Yash, Harsh, Abhishek, Akrit, Aroma, and several others, for making grad school enjoyable. And to older friends – Prateek, Vineet, Shalaka, Amulya, Aastha, Rohan, Sahil – for making me look forward to trips back home.

Finally, I am tremendously grateful to my parents, and to Shruti, for their unconditional love and support that have kept me going through it all. Thank you for all that you do.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In this thesis, we consider the problem of clinical image classification for the purpose of aiding doctors in dermatological disease diagnosis. Diagnosis of dermatological disease condition from images poses two major challenges for standard off-the-shelf techniques: First, the distribution of real-world dermatological datasets is typically long-tailed. Second, intra-class variability is large. To address the first issue, we formulate the problem as low-shot learning, where once deployed, a base classifier can rapidly generalize to diagnose novel conditions given very few labeled examples. To model intra-class variability effectively, we propose Prototypical Clustering Networks (PCN), an extension to Prototypical Networks [1] that learns a mixture of "prototypes" for each class. Prototypes are initialized for each class via clustering and refined via an online update scheme. Classification is performed by measuring similarity to a weighted combination of prototypes within a class, where the weights are the inferred cluster responsibilities. We demonstrate the strengths of our approach in effective diagnosis on a realistic dataset of dermatological conditions. Further, we demonstrate the generality of our approach by applying it to the standard *mini*ImageNet benchmark in few-shot learning, showing improved performance on the generalized few-shot setting.

CHAPTER 1
INTRODUCTION

Access, equity, quality, and cost-effectiveness are key issues facing health care across the world [2]. Telemedicine (literally, 'healing at a distance" [3]) is increasingly becoming effective in reducing the friction in delivering accessible quality health care across the globe *c*.f. [4, 5, 6]. Such a rapid influx of telemedicine is only possible due to exponential progress in information and communication technologies, in particular expanding the scope of telemedicine to encompass internet-based applications with rich multimedia content [2].

To truly scale the world's best healthcare to every human being, these internet based medical applications need to *help doctors scale*; In addition to being a bridge to connect patients to doctors at a distance, telemedicine applications need to *aid* doctors at different levels of decision-making; be it triaging the patient to the right doctor at the right time or aiding the doctors in diagnosis. The focus of this paper is in scaling doctors in the context of a key health service, namely, dermatology. Globally, skin disease is one of the most common human illnesses that affects 30% to 70% of individuals, with even higher rates in at-risk subpopulations where access to care is scarce [7, 8, 9, 10, 11]. Untreated or mistreated skin conditions often lead to detrimental effects including physical disability and death [11].

A large fraction of skin conditions are diagnosed and treated at the first point of contact, *i.e.* by primary care practitioners (PCPs), either in a clinical setting or in a telemedicine scenario. While this makes access to care faster, recent studies indicate that general physicians, especially those with limited experience, may not be well-trained for diagnosing many skin conditions [12, 13]. In addition, people with no or little access to health care systems often depend on their own search and 'image recognition capabilities' to self (mis-)diagnose and treat. While there is a recent surge in online services for closing the gap of healthcare access, these services also have similar problems [14]. The need to find effective solutions to

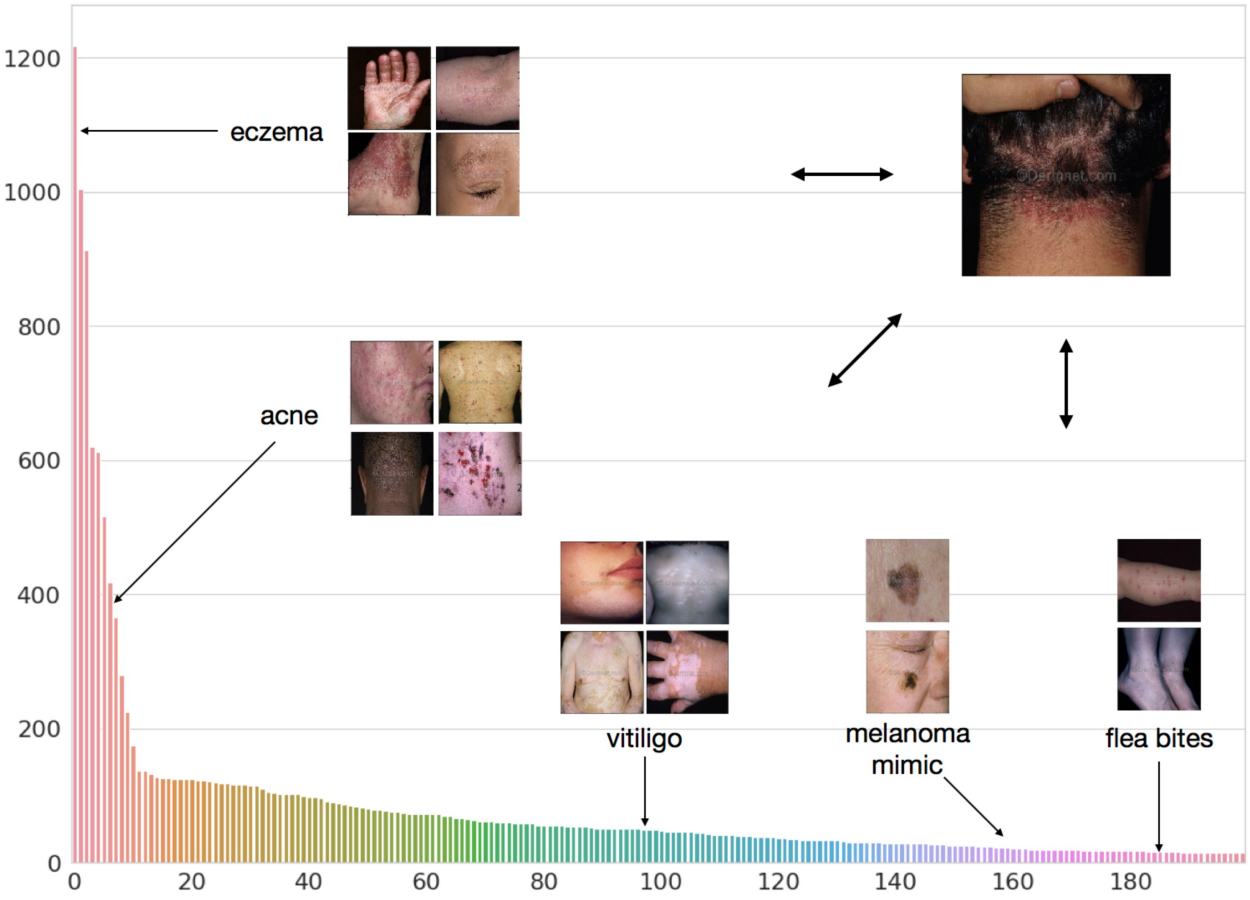

Figure 1.1: Long-tailed class distribution of Dermnet (shown here for the top-200 classes). Also shown are nearest neighbors to four of the many prototypes learned for select classes using the proposed Prototypical Clustering Network approach. This is illustrative of the huge intra-class variability in the data. For a novel test image, shown at the upper right corner, the model predicts the correct class by measuring weighted similarity to per-class clusters in the embedding space learned through a deep convolutional neural network.

aid primary care physicians in accurate diagnosis motivates this work.

Why is diagnosis of skin conditions hard for doctors? One important factor is the sheer number of dermatological conditions. The International Classification for human diseases [15] enumerates more than 1000 skin or skin-related illnesses. However, most PCPs are trained on a few tens of common skin ailments [16] under the assumption that this will enable accurate diagnoses in most cases. Recent studies indicate flaws in this assumption [16]. To make an accurate diagnosis, the knowledge of all possible diseases is important, especially to workup and eliminate possible life-threatening conditions. The difficulty of diagnosis is further com-

pounded by large intra-class variability (eg. acne may occur on the face, hand, scalp, etc.). To motivate the scale of this problem, see Figure 1.1, where we show the class distribution of Dermnet Skin Disease Atlas[1], a publicly available large-scale dataset of dermatological conditions. The plot shows examples illustrating the intra-class variability found in the dataset. This makes accurate diagnosis challenging even for experienced dermatologists.

These issues create an opportunity for incorporating machine learning systems into the doctor's workflow, aiding them in sieving through possible skin conditions. AI systems have shown promising results in the healthcare domain, with early applications on automated detection of skin lesions from images, *c.f.* [17] and diagnosis based on radiology data *c.f.*[18]. Inspired by these recent successes, this thesis tackles the problem of fine-grained skin disease classification. We conjecture that a high fidelity AI system can serve as a diagnostic decision support system to general physicians. By suggesting candidate diagnoses, it can greatly reduce effort and compensate for the possible lack of experience or time at the point of care. In the context of teledermatology with a store-and-forward approach that involves asynchronous evaluation by dermatologists, such a system can aid in triaging the right doctor resource in a timely manner, especially when acute conditions need immediate care [13]. This would positively impact many patients that have acute skin conditions that need to be dealt within a couple days, and such triaging functionality can greatly improve the time to care.

To this end, we pose dermatological image classification as a *few-shot learning* problem, where a base classifier, once deployed, needs to be easily extensible to new classes from a few labeled examples (potentially labeled by a physician). Our approach pursues the following objectives:

- **Modeling intra-class variability.** Several conditions contain significant intra-class variability, *e.g.* a condition like acne may occur on the face, back, scalp, etc.

- **Learning without forgetting:** The ability to diagnose novel conditions must not compromise the performance on base classes.

[1] http://www.dermnet.com/

- **Privacy Preservation:** As access to most dermatological data (usually part of Electronic Health Records) is strictly controlled, the model should not require access to the original (potentially, proprietary) training data when being extended.

Our proposed model, that we call Prototypical Clustering Networks (PCN), extends prior work on Prototypical Networks [1] to represent a class as a mixture of prototypes instead of a single prototype. Training this classifier involves learning an embedding space while simultaneously learning to represent each class as a mixture of prototypes. Prototypes are initialized for each class via clustering and refined via an online update scheme. Classification is performed by measuring similarity to a weighted combination of prototypes within a class, where the weights are the inferred cluster responsibilities. The examples shown in Figure 1.1 are, in fact, nearest neighbors to prototypes of the classes learned using the proposed approach. Quantitative results demonstrate the strengths of our approach for dermatological disease diagnosis.

To summarize, in this thesis we make the following contributions:

- To deal with the long tail of automated dermatological diagnosis from clinical images, we pose it as a few-shot learning problem, and design a benchmark and metrics to measure progress.

- We propose Prototypical Clustering Networks, an extension to prior work in Prototypical Networks [1] that learns a mixture of prototypes for each class to effectively model multimodal class distributions.

- We analyze the effectiveness of our approach on our proposed benchmark against several baselines, and further study the generality of our approach by applying it to the standard *mini*ImageNet few-shot learning benchmark, demonstrating performance competitive with the state of the art.

CHAPTER 2
RELATED WORK

2.1 Dermatological Classification.

A few prior works address the problem of dermatological classification. In [17], authors focus specifically on diagnosing skin cancer, and establish a benchmark on a large closed-source dataset of skin lesions by finetuning a pretrained deep convolutional neural network (CNN). In [19], authors study the problem of skin disease diagnosis on the Dermnet dataset but focus on coarse 23-way classification using its top-level hierarchy. In [20], the authors propose a benchmark dataset for skin disease diagnosis containing 6584 clinical images. In follow up work, [21] propose an approach to learn representations inspired by diagnostic criteria employed by dermatologists on this dataset. In this work, we study fine-grained recognition of skin conditions on the Dermnet dataset which is a signficantly larger dermatological resource containing over 23000 images. Further, we formulate this as a few-shot learning setup, and propose a method to model multimodal classes and generalize effectively to previously unseen novel classes with very little labeled data.

2.2 Class-imbalanced datasets.

Real-world visual datasets frequently possess long tails [22, 23, 24], and learning robust representations from such data is a topic of active research. Conventional training methods typically lead to poor generalization on tail classes as class-prior statistics are skewed towards the head of the distribution. Simple techniques such as random oversampling (or undersampling) by repeating (or removing) tail instances are found to help mitigate this issue to a degree [25]. Alternative approaches perform meta learning to transfer knowledge

from data-rich head classes to the tail [23]. Recently, [26] propose learning a discriminative embedding in which a Gaussian mixture model is used to balance class-priors and lead to better generalization. In this work, we propose a few-shot learning approach on a real-world imbalanced dataset of dermatological conditions, and demonstrate strong few-shot generalization capabilities.

2.3 Few-shot learning.

Few-shot learning aims to learn good class representations given very few training examples [1, 27, 28, 29, 30]. Main paradigms of approaches include simulating data starved environments at training time, and including non-parametric structures in the model as regularizers. Matching networks [27] learn an attention mechanism over support set labels to predict query set labels for novel classes. Prototypical networks [1] jointly learn an embedding and centroid representations (as class *prototypes*), that are used to classify novel examples based on Euclidean distance. In both [27] and [1], embeddings are learned end-to-end and training employs episodic sampling. Some recent approaches learn to directly predict weights for new layers from embedding layer activations of support examples [31, 32]. In [33], the motivation is to perform few shot learning 'without forgetting', i.e. extending to novel classes without catastrophic forgetting (also studied as generalized few-shot learning in [34]). In an incremental few-shot learning context, [35] propose a class-centroid based representation in an embedding space learned using a generative model. In [29], authors study few-shot learning by creating an imbalanced few-shot benchmark from ImageNet [36], and propose a method to "hallucinate" additional samples for such data-starved classes. In this work, we focus on a similar setup on the real-world long-tailed dermatological dataset. We propose an extension to [1] to model the multimodal nature of diverse classes, and demonstrate how this also helps generalize better to data-starved novel classes.

2.4 Prototypical Networks.

Prior extensions to Protoypical Networks exist in the literature, and here we distinguish our contributions [28, 37]. In [28], authors propose extending Prototypical Networks to a semi-supervised setting by using unlabeled examples while producing prototypes. In [37], authors propose additionally predicting a covariance estimate for each embedding and using a direction and class dependent distance metric instead of euclidean distance. Yet others have looked at employing prototypical networks for active few-shot learning [38]. In this work, we extend prototypical networks to model multimodal classes in an automated diagnostic setting by learning multiple prototypes per class, that are initialized via clustering and refined via an online update scheme.

CHAPTER 3

APPROACH

We formulate dermatological image classification as a low-shot learning problem. During training time, we have access to a labeled dataset of images $S = \{(x_1, y_1), ..., (x_N, y_N)\}$ where each x_i is an observation and $y_i \in \{1, ..., K_{base}\}$ is the label mapping to one of the *base* classes known at training time. At test time, we are also provided with a small labeled dataset corresponding to K_{novel} novel classes, and must learn to perform $K_{base+novel}$ way classification.

Algorithm 1 Training episode loss computation for Prototypical Clustering Networks. N is the number of examples in the training set, K_{base} is the number of base classes for training, M_k is the number of clusters for class k, $N_C \leq K_{base}$ is the number of classes per episode, N_S is the number of support examples per class, N_Q is the number of query examples per class. RANDOMSAMPLE(S, N) denotes a set of N elements chosen uniformly at random from set S, without replacement. *Differences from Algorithm 1 in [1] in blue*

Input:Training set $\mathcal{D} = \{(x_1, y_1), \cdots, (x_N, y_N)\}$, where each $y_i \in \{1, \cdots, K\}$. $\mathcal{D}_k$ denotes the subset of $\mathcal{D}$ containing all class prototypes, i.e. elements $(x_i, y_i) = \{\mu_{z,k}\}_{z=1}^{M_k} \forall k \in \{1, \cdots, K\}$

Output: The loss J for a randomly generated training episode

V $\leftarrow$ RANDOMSAMPLE($\{1, \cdots, K\}$, N_C) ▷ Select class indices for episode

for $k \in \{1, \cdots, N_C\}$ **do**

  $S_k \leftarrow$ RANDOMSAMPLE($\mathcal{D}_{v_k}$, N_S) ▷ Select support examples

  $Q_k \leftarrow$ RANDOMSAMPLE($\mathcal{D}_{v_k \setminus S_k}$, N_Q) ▷ Select query examples

  for $(x, y) \in S_k$ **do** ▷ Compute probabilistic assignment of x to y's clusters

    $q(z|k, x) = \frac{\exp(-d(f_\phi(x), \mu_{z,k})/\tau)}{\sum_{z'} \exp(-d(f_\phi(x), \mu_{z',k})/\tau)}$

  for $z \in \{1, \cdots, M_k\}$ **do**

    $\mu_{z,k}^{new} \leftarrow \alpha \mu_{z,k}^{old} + (1 - \alpha) \frac{\sum_{(x,y) \in S_k} q(z|k,x) f_\phi(x)}{\sum_{(x,y) \in S_k} q(z|k,x)}$

$L_\phi \leftarrow 0$

for $k \in \{1, \cdots, N_C\}$ **do**

  for $(x, y) \in Q_k$ **do**

    $L_\phi \leftarrow L_\phi + \frac{1}{N_C N_Q} \left[\sum_z q(z|k, x) d(f_\phi(x), \mu_{z,k}) + \log \sum_{k'} \exp(- \sum_{z'} q(z'|k', x) d(f_\phi(x), \mu_{z',k'})) \right]$

3.1 Model

Prototypical Clustering Networks (PCN) builds upon recent work in Prototypical Networks [1]. PCN represents each class using a set of prototypical representations learned from the data. Let $\{\mu_{z,k}\}_{z=1}^{M_k}$ be the collection of M_k prototypes for class k. Then, at test time, we measure similarity to these representations to derive its corresponding class label. In particular,

$$p(y = k|\mathrm{x}) = \frac{\exp(-\sum_z q(z|k,x)d(f_\phi(x), \mu_{z,k}))}{\sum_{k'} \exp(-\sum_z q(z|k,x)d(f_\phi(x), \mu_{z,k'}))} \tag{3.1}$$

where $f_\phi(x)$ is the embedding function with learnable parameters ϕ that maps input x to a learned representation space, d is a distance function and $q(z|k,x)$ (eq. 3.2) is soft assignment of examples to clusters from the class. When $M_k = 1$ for all classes, we revert to prototypical networks.

3.2 Model training

The goal is to learn a model with parameters ϕ so as to maximize the probability of the correct class such that $\phi^* = \arg\max_\phi \sum_{(\mathrm{x},y)} \log p(y|\mathrm{x};\phi) = \arg\min_\phi L_\phi$, where L_ϕ is the corresponding loss function. We use episodic training [1, 27, 39] to learn the embedding function by optimizing the loss and updating the cluster prototypes for each class. In particular, a training epoch consists of E episodes. Algo. 1 provides the details of computing the loss for one episode that is used in learning the function. We describe key components of the algorithm below:

3.2.1 Class-specific cluster responsibilities:

The assignment of an example within each class is given by:

$$q(z|k,x) = \frac{\exp(-d(f_\phi(x), \mu_{z,k})/\tau)}{\sum_{z'} \exp(-d(f_\phi(x), \mu_{z',k})/\tau)}, \tag{3.2}$$

where τ is temperature parameter that controls the variance of the distribution. As we decrease the temperature, the distribution becomes more peaky, and becomes flatter as we increase it. The importance of τ can be understood by studying the loss function L_ϕ in line 15 of Algo. 1. During training, if clusters are well-separated, $q(z|k,x)$ will be peaky so that each example effectively contributes to the update of a single cluster in a class, whereas if clusters overlap, $q(z|k,x)$ will be diffuse and the corresponding example will contribute to multiple prototypes. Therefore, during training, we typically set τ to favor peaky distributions so that learned clusters focus on different regions of the input space.

3.2.2 Class-specific cluster prototypes:

In episodic training, an epoch corresponds to a fixed number of episodes and within each episode, classes are sampled uniformly. In our setting with huge class imbalances, this translates to examples from tail classes being oversampled, while examples from the head may be undersampled within an epoch. This can adversely affect model training. To mitigate this, at the start of an epoch, we initialize cluster prototypes for each class using k-means [1] on the learned embedding representation of examples from the entire training set of that class. We rerun this clustering step at the start of each epoch to prevent collapse to using only a single cluster per class.

Subsequently, in each episode, we use an *online* update scheme that balances between the local estimate of the prototype computed from embeddings of the current support set (to account for the evolving embedding space), and the prototypes learned so far:

$$\mu_{z,k}^{new} \leftarrow \alpha \mu_{z,k}^{old} + (1-\alpha)\frac{\sum_{(x,y)\in S_k} q(z|k,x) f_\phi(x)}{\sum_{(x,y)\in S_k} q(z|k,x)}, \tag{3.3}$$

[1] We empirically found using a fixed number of clusters per class to work best. Choosing an optimal number of clusters per class is a difficult problem, largely due to differences in the amount of intra-class variability across classes. In an attempt to bypass this challenge, we experimented with using affinity propagation[40], but found it to be highly unstable as a) the similarity function (needed for clustering) is based on the embedding representation that is optimized concurrently, and b) there is no straightforward online update rule (equivalent to eq. 3.3) to subsequently update within an episode.

where α trades off memory from previous episodes and its current estimate.

3.2.3 Understanding the role of multiple clusters

We can derive insights about the role of multiple clusters by interpreting PCN as a non-linear generalization of PN ([1]). Using squared Euclidean distance in Eqn. 3.1, we expand the term in the exponent so that:

$$\begin{aligned} -\sum_z q(z|k,x)||f_\phi(x) - \mu_{z,k}||^2 &= \text{const. for k+} \\ 2\sum_z q(z|k,x) f_\phi(x)^T \mu_{z,k} - \sum_z q(z|k,x)\mu_{z,c}^T \mu_{z,k} & \\ = \text{const. for k} + 2w_{k,x}^T f_\phi(x) - b_k & \end{aligned} \tag{3.4}$$

where

$$w_{k,x} = \sum_z q(z|k,x)\mu_{z,k} \tag{3.5}$$

$$b_{k,x} = \sum_z q(z|k,x)\mu_{z,k}^T \mu_{z,k} \tag{3.6}$$

The last two terms in Eqn. 3.4 are non-linear functions of the data, where the non-linearity is captured through both the embedding and the mixing variables. The functional forms of the factors, namely $w_{k,x}$ and $b_{k,x}$, also sheds light on the advantage of using multiple clusters per class. In particular, unlike in prototypical networks, $w_{k,x}$ is an *example-specific* "prototypical" representation for class k, obtained by using a convex combination of prototypes for the class, weighted by posterior probability over within-class cluster assignments. When $q(z|k,x)$ is confident with a peaky posterior, the model behaves like a regular prototypical network. In contrast, when the posterior has uncertainty, PCN interpolates between the prototypes by modulating $q(z|k,x)$.

CHAPTER 4

RESULTS

4.1 Experimental setup

In this section we provide details of our experimental setup and results.

4.1.1 Dataset:

We construct our dataset from the Dermnet Skin Disease Atlas, one of the largest public photo dermatology sources containing over 23,000 images of dermatological conditions. These images are clinical images collected through various sources, including mobile phones, digital cameras, etc. and so vary in pose, lighting, and resolution. Images are annotated at a two level hierarchy – a coarse top-level containing parent 23 categories, and a fine-grained bottom-level containing more than 600 skin conditions. We focus on the more challenging bottom-level hierarchy for our experiments. First, we remove duplicates from the dataset based on name, and also based on collisions found using perceptual image hashing [41].

Figure 1.1 presents a histogram of the resulting class distribution, filtered to the top-200 classes. We can see that the dataset has a long tail with only the 100 largest classes having more than 50 images; beyond 200 classes, the number of images per class reduces to double digits, and with 300 classes to single digits. Unless otherwise stated, for experimental comparisons, we focus on the top-200 classes so that $K_{base+novel} = 200$, which contains 15507 images. Similar to [29], we treat the largest 150 classes as base classes ($K_{base} = 150$) and the remaining 50 classes as novel ($K_{novel} = 50$). This helps in ensuring reasonably sized splits for training, validation and evaluation. In particular, we sample $max(5, 20\%)$ without replacement for each base class to get validation and test splits (3163 images each). The remaining is used for training (9181 images). For the low-shot learning phase, following the

procedure used in [29], we sample 5 examples each for training and testing, respectively. We report mean and standard deviation of metrics over 10 cross validation runs.

4.1.2 Metrics:

As our dataset is imbalanced, we report mean of per-class accuracy (mca) as our metric, treating each class as equally important. For a dataset consisting of C classes, with T_c examples in each class, mean accuracy is the average of per-class accuracies: mca = $\frac{1}{C}\sum_c \frac{\sum_{t=1}^{T_c} I[\hat{y}^{(t)}[0]=y^{(t)}]}{T_k}$, where, for t^{th} example, $\hat{y}^{(t)}[j]$ is the j^{th} top class predicted from a model and $y^{(t)}$ is its corresponding ground truth label, where I denotes the indicator function.

We use $\text{mca}_{\text{base+novel}}$ to report combined mca performance of examples from all classes. mca_{base} corresponds to evaluation of classifier on $K_{base+novel}$ classes but restricted to only test examples from base classes. Similarly, $\text{mca}_{\text{novel}}$ corresponds to evaluation of test examples in novel classes while performing $K_{base+novel}$ way classification.

We also report recall@k (k $\in \{5, 10\}$). This metric (also called *sensitivity*) is valuable in deployment contexts that involve aiding doctors in diagnosis, as it ensures that the relevant disease condition is considered within a small range of false positives. However, since our test set is imbalanced, recall@k metrics unfairly reward strong performance on the head classes, and so to provide a fairer comparison, we report *balanced* (or macro) recall@k metrics, wherein we compute recall@k for each class and average, treating each as equally important.

4.1.3 Model:

We initialize a 50-layer ResNet-v2 [42], a state-of-the-art convolutional neural network architecture for image classification, with ImageNet pretraining [1], and train a Prototypical Clustering Network as described in Sec. 3 on K_{base} classes. We use 10 and 4 clusters per class for base and novel classes respectively, and a temperature of 1.0 (all picked via grid search).

[1] We also experimented with training from scratch, and found it to perform significantly worse

4.1.4 Baselines

1. **Prototypical Network (PN):** We train an ImageNet-pretrained ResNet-V2 CNN as a Prototypical Network [1] on K_{base} classes.
2. **Finetuned Resnet with nearest neighbor (FT_K-*NN)**: Here, we finetune an ImageNet-pretrained ResNet-v2 convolutional neural network with 50 layers [42] on training data from K classes. We report numbers for K $\in \{K_{base}, K_{base+novel}\}$. The model is trained as a softmax classifier with a standard cross entropy objective. Then, we obtain embeddings for the entire training set consisting of $K_{base+novel}$ classes. This is used to perform $*$-nearest neighbor classification on the test set from all of $K_{base+novel}$ classes.
3. **Finetuned ResNet (FT_K-CE)**: We use the same ResNet model as the above, with $K = K_{base+novel}$, i.e. trained for $K_{base+novel}$ way classification using training data from both base and novel classes, and validated using the corresponding validation set on the base classes (due to lack of data in novel classes). We train the model with class balancing. This is a strong baseline as we use all $K_{base+novel}$ during training, and also due to class balancing, which has been shown to be a reliable strategy for dealing with class imbalance when training CNN models [25].

Hyperparameters: For PN and PCN, we use episodic batching with 10-way 10-shot classification (at train), and 200 episodes per epoch. At test, we compute per-class prototypes using the training set for all $K_{base+novel}$ classes, and perform $K_{base+novel}$ way classification. The embedding function for PCN and PN produces 256-dimensional embeddings, and uses the same architecture as in FT_K-CE (with one less fully connected layer). Models are trained with early stopping using Adam [43], a learning rate of 10^{-4}, and L2 weight decay of 10^{-5}.

4.1.5 Main results

Table 4.1 highlights our main MCA results. The table shows test set MCA over the 200 classes available during test time for two different low shot settings: train shots of 5 and 10

Table 4.1: Mean per-class accuracy (MCA) on top 200 classes. We focus on the low-shot setting, using all training data for the base classes (the largest 150) and $n = 5$ or 10 examples for the remaining 50 classes (denoted as "novel"). Note that FT_{200}-CE and FT_{200}-*NN use training data for all 200 classes, whereas the other models use only the base classes for representation learning, using support sets from the remaining 50 classes after training to derive prototypes.

	$n = 5$			$n = 10$		
Approach	$\text{mca}_{\text{base+novel}}$	mca_{base}	$\text{mca}_{\text{novel}}$	$\text{mca}_{\text{base+novel}}$	mca_{base}	$\text{mca}_{\text{novel}}$
FT_{150}-1NN	46.2 ± 0.8	55.3 ± 0.3	18.8 ± 3.3	49.5 ± 0.3	54.9 ± 0.5	33.4 ± 1.4
FT_{150}-3NN	44.3 ± 0.3	54.8 ± 0.5	12.8 ± 1.5	47.0 ± 0.6	54.1 ± 0.4	25.6 ± 1.5
FT_{200}-1NN	46.5 ± 0.4	54.2 ± 0.3	22.5 ± 0.8	49.9 ± 0.5	53.8 ± 0.4	38.3 ± 1.3
FT_{200}-3NN	44.7 ± 0.4	52.6 ± 0.2	20.9 ± 2.0	48.0 ± 0.1	52.5 ± 0.1	34.3 ± 0.2
FT_{200}-CE	**47.8 ± 0.5**	**55.8 ± 0.7**	24.0 ± 3.2	**51.5 ± 0.4**	**55.2 ± 0.3**	40.4 ± 2.4
PN	43.9± 0.4	48.7 ± 0.4	29.6 ± 2.4	44.9 ± 0.8	47.6 ± 0.4	37.1 ± 3.4
PCN (ours)	**47.8 ± 0.7**	53.7 ± 0.2	**30.0 ± 2.8**	**50.9 ± 0.6**	51.4 ± 0.3	**49.6 ± 2.8**

with test set of 5. In both low shot settings, we observe the following trends:

- FT_K-CE and PCN shares similar performance on combined MCA. However, their performance on the base and novel classes are quite distinct. Much of the performance gains for FT_K-CE come from the base classes that have a lot more training examples than novel classes. In contrast, PCN, through episodic training aims at learning discriminative feature representations that are generalizable to novel classes with highly constrained number of examples; this is evident by its significantly better performance (9% absolute gains) in generalizing to novel classes. At the same time, PCN ensures that performance on novel classes doesn't come at the cost of lower accuracy on the base classes. Also note that the FT_K-CE model requires re-training for adding novel classes while PCN only requires a single forward pass to learn prototypes for novel classes.

- FT_K-*NN models learn robust representations for base classes, but are unable to generalize to novel classes, outperforming a regular PN model on top-200 MCA but underperforming against PCN. Interestingly, we find that increasing the number of nearest

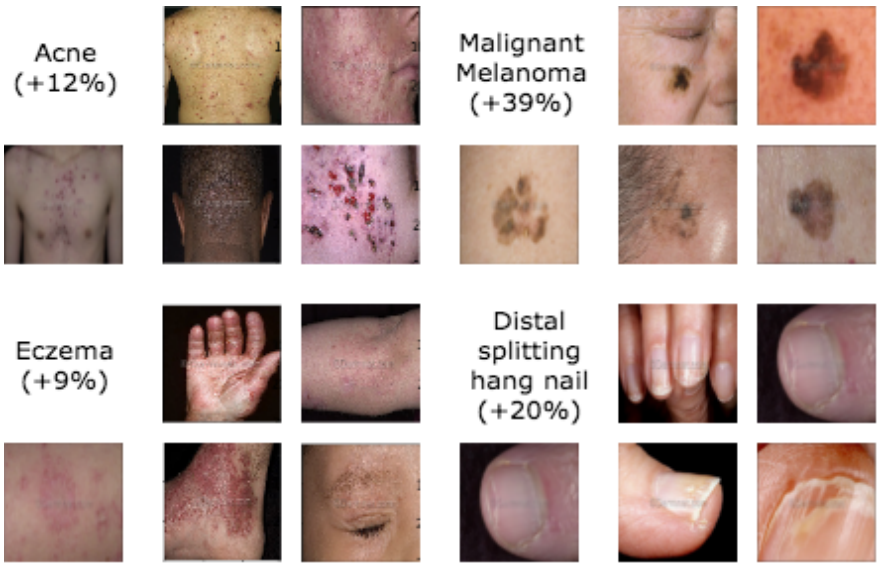

Figure 4.1: Learned prototypes are shown using their nearest neighbors in the training set. Each skin condition is in a 2×3 grid; The image below the name of the skin condition corresponds to PN while the 2×2 grid corresponds to nearest neighbors of four cluster prototypes. +X% below the name denotes improvement of PNC over PN for that class for $\text{mca}_{base+novel}$,. Note that novel classes such as 'Distal splitting hang nail' can also be diverse, as shown by clusters identified with PCN.

neighbors leads to poor performance, especially on novel classes. This could be due to sparsity of training data.

- PCN outperforms PN on combined base and novel classes by a large margin. This demonstrates that representing classes with multiple prototypes leads to better generalization on both base and novel classes. In Figure 4.1, we show the nearest neighbor to class prototype for PN and to four of the PCN prototypes, for select classes. We can see that PCN has learned to model intra-class variability much more effectively. As an example, for eczema and acne classes we can see that PCN learns clusters corresponding to these skin conditions in different anatomical regions. We provide a more in-depth comparison in the next section.

In Table 4.2 we report balanced recall@k (br@k) performance. Here we clearly find PCN to outperform all baselines owing to strong performance across the board on base and novel classes.

Table 4.2: Balanced Recall@k on top 200 classes.

	Approach	br@5$_{\text{base+novel}}$	br@5$_{\text{base}}$	br@5$_{\text{novel}}$	br@10$_{\text{base+novel}}$	br@10$_{\text{base}}$	br@10$_{\text{novel}}$
n=5	FT_{200}-CE	65.4 ± 0.7	74.6 ± 0.2	38.0 ± 2.1	73.1 ± 0.5	82.3 ± 0.2	45.3 ± 1.5
	PN	66.5 ± 0.6	67.6 ± 0.2	**63.2 ± 2.3**	75.3 ± 0.5	74.9 ± 0.1	**76.3 ± 2.1**
	PCN (ours)	**70.7 ± 0.6**	**74.5 ± 0.2**	**59.2 ± 2.6**	**79.1 ± 1.0**	**81.4 ± 0.3**	**72.2 ± 3.6**
n=10	FT_{200}-CE	69.9 ± 0.5	**73.8 ± 0.6**	58.0 ± 3.2	77.9 ± 0.6	81.9 ± 0.2	65.9 ± 2.6
	PN	67.5 ± 0.4	66.0 ± 0.3	**72.2 ± 1.7**	75.9 ± 0.6	72.9 ± 0.3	**84.7 ± 2.2**
	PCN (ours)	**71.4 ± 0.7**	70.4 ± 0.2	**74.5 ± 2.6**	**79.9 ± 0.5**	**78.2 ± 0.2**	**85.0 ± 2.0**

Table 4.3: Does post-hoc clustering on PN help? (CPC denotes number of clusters per class)

Model	Eval CPC (base / novel)	mca$_{\text{base+novel}}$	mca$_{\text{base}}$	mca$_{\text{novel}}$	recall@5	recall@10
PN	1 / 1	43.9 ± 0.4	48.7 ± 0.4	29.6 ± 2.4	70.9 ± 0.4	80.2 ± 0.3
PN	1 / 4	44.4 ± 0.5	50.4 ± 0.4	26.4 ± 2.3	74.2 ± 0.2	83.5 ± 0.3
PN	10 / 4	43.8 ± 0.8	50.3 ± 0.2	24.2 ± 3.0	75.6 ± 0.2	84.0 ± 0.2
PCN (ours)	10 / 4	**47.8 ± 0.7**	**53.7 ± 0.2**	**30.0 ± 2.8**	**77.8 ± 0.2**	**86.0 ± 0.4**

4.1.6 Comparison between PCN and PN

PCN or PN with post-hoc clustering? To understand the effectiveness of PCN, we compare it to a PN model in which we perform "post-hoc" clustering: (a) cluster novel class representations using the PN model's learned embeddings (with cluster size of 4) (b) cluster both base and novel class representations (with cluster size of 10 and 4, as in PCN). Rows 1-3 in Table 4.3 compare the performance between different post-hoc clustering variants of PN against PCN. We see that PCN leads in all metrics across the board; thus such post-hoc clustering does not lead to improved performance. A reason for this is that the PN model is optimized to learn representations assuming a projection to a single cluster for each class, and hence clustering on such learned representation does not improve performance. This further validates the importance of training with multiple clusters.

Varying test shot: Table 4.4 highlights the effect of number of support examples (shot). As we increase the shot, the performance improves on both methods, but that improvement is larger for PCN than for PN. Because of this, the performance gap between the two methods drastically increases. PCN is better at utilizing the availability of more data by partitioning

the space with clusters.

Extending the tail: In this experiment, we study the performance as we vary the number of novel classes at test time from 50 to 150, bringing the total number of classes up from 200 to 300. Table 4.4 provides the comparison. We use a train and test shot of 2 and 5 respectively since most classes in these additional in the long tail of the dataset are extremely sparse. Results are reported with 10-fold cross validation. While there is a drop in performance for both models due to small shot sizes, we can see that the performance gap between PCN and PN continues to hold.

Table 4.4: Extending Shot, Tail.

		PN	PCN
Shot	2	42.26	45.36
	5	44.35	47.79
	10	44.93	50.92
K	200	42.26	45.58
	250	37.08	40.37
	300	34.70	37.67

As a sanity check to ensure the general applicability of our approach, as well as to compare our approach to other state of the art few-shot learning approaches, we also conduct experiments on the *mini*ImageNet dataset [27], which is an established benchmark in the few shot learning literature. We find that our approach performs competitively with the state of the art.

4.1.7 Qualitative Results

Figure 4.2 provides qualitative examples comparing the three methods. For each query image, the figure shows the predicted labels for the three methods. For PN, the figure shows the nearest neighbor to the prototype of the predicted class, while for PCN,the figure shows the nearest neighbor to the closest cluster prototype of the predicted class. Consider column 1. Acne is one of the largest classes in the base classes with large intra-class variability. Both FT_K-CE and PCN can diagnose this example correctly. However, PN due to its limited capac-

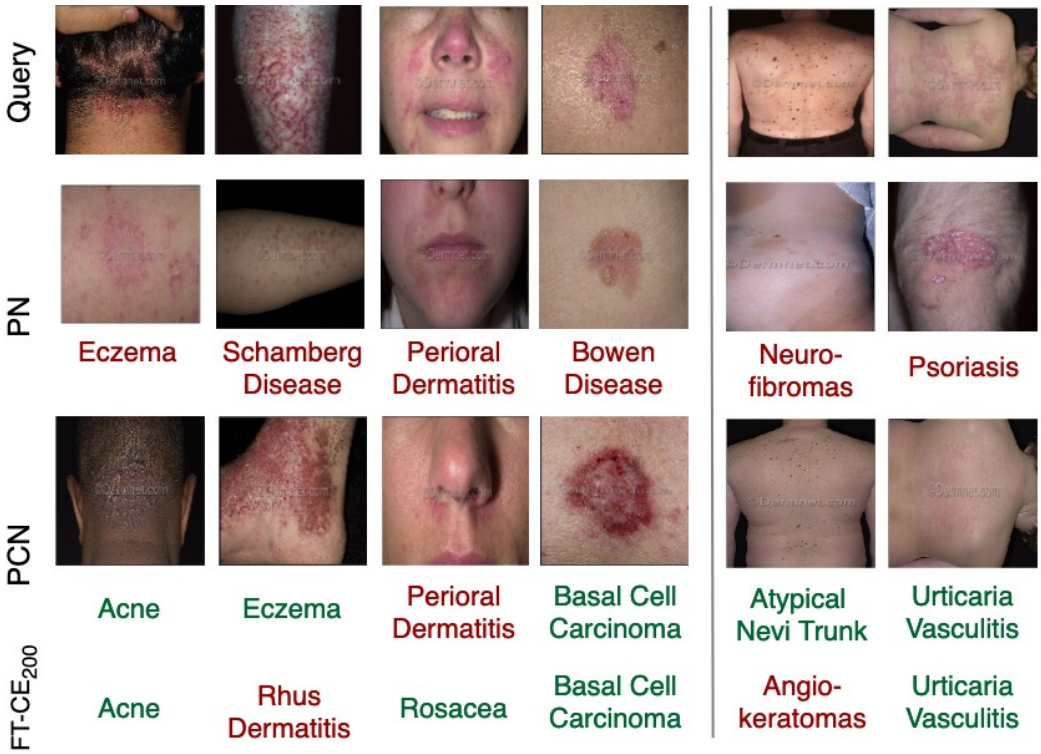

Figure 4.2: For each query image in the test set, we compare PCN with PN and FT_{200}-CE. For each image, we color code correct label with green and incorrect with red. For PN, we show the nearest neighbor to the prototype of the *predicted* class. For PCN, we show the nearest neighbor of the top cluster according to $q(z|c, x)$ of the predicted class. The last two columns correspond to examples from novel classes.

ity to represent the huge variability in the class is confused with another large class, namely, eczema. PCN, due to having access to multiple clusters can learn a better representation and correctly diagnose acne.

In column 4, we present a case in which both PN and FT_K-CE identified the query image as atypical nevi dermoscopy, while PCN correctly classified it as malignant melanoma. Atypical nevi are 'funny-looking' moles that are precursors to melanoma. It has been recently studied that dermoscopic features discriminating between atypical naevi and melanoma require expert interpretation through longitudinal monitoring, but are often ignored as simple moles [44]. In contrast, consider Column 7: FT-CE misdiagnose atypical nevi as Angiokeratoma, a benign skin lesion of capillaries, resulting in small marks of red to blue color. In the data-starved setting, FT_K-CE and PN are unable to differentiate the two skin conditions

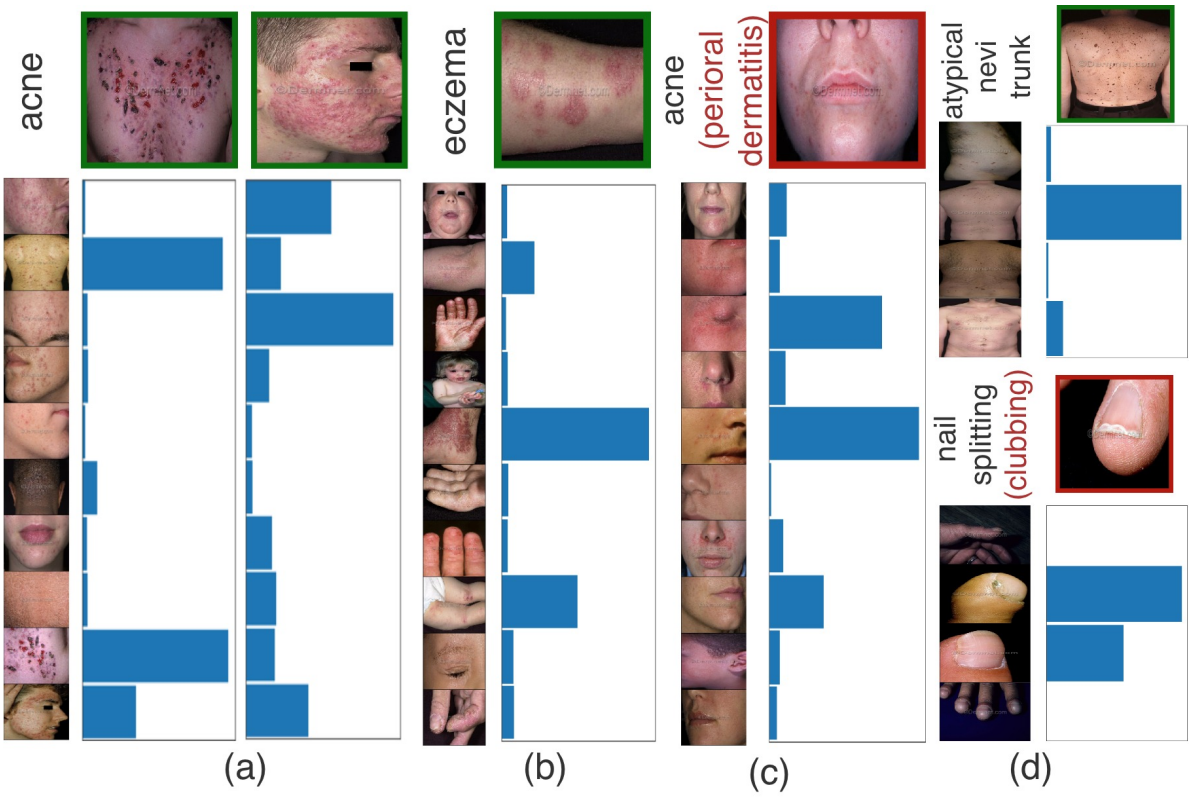

Figure 4.3: Effectiveness of using multiple clusters. Shown for base and novel classes. (a)-(b): Examples from test set that are correctly classified by PCN. For each class, we show the nearest neighbor to the learned prototypes. We also present examples (columns) whose labels are correctly predicted and the inferred cluster responsibilities $q(z|c, x)$ conditioned on the correct class. (c): Test set example that is misclassified by PCN. Correct label is shown in black, while the incorrect prediction is shown in red. We show the nearest neighbors to the learned cluster prototypes of the *predicted* (incorrect) class, and the corresponding cluster responsibilities. Note that green outlines around query images denote correct classification while red denotes incorrect classification.

while PCN can better match up to the support set.

Effectiveness of multiple clusters. In Sec 3, we show how PCN can interpolate between the learned prototypes by modulating $q(z|c, x)$. In Figures 4.3 and 4.4 we show some qualitative examples to illustrate this. We show query images from the test set for various classes, with a mix of correct and incorrectly classified examples. Below the class label, we show the nearest neighbor image from the training set to each of the learned prototypes for the class *predicted* by PCN, and below each query image, cluster responsibilities placed by the model on each of these prototypes. As an example, consider Figure 4.3(a). For this class (acne), we

show an example that is correctly classified by PCN. Interestingly, 4.3(a), the model appears to interpolate between two prototypes that are similar to the query image in pose and skin texture respectively, to make a correct prediction. We can see that the $q(z|c, x)$ distribution varies quite a bit across examples, being a lot more diffuse in some cases than others. It can also be seen that the model learns to accurate place probability mass on similar prototypes. Similarly for 4.3(b) (eczema), the model is accurately able to identify eczemas on the face, arm, and hand, by combining the most relevant prototypes. While these classes see relatively diffuse responsibility distributions, the distribution is far more peaked for the varicella class in (c). Figure 4.3(c), (d)(bottom) shows incorrectly predicted examples; Even for these examples, the model seems to interpolate, albeit incorrectly, to make predictions. Note that Figure 4.3(d) corresponds to examples from novel classes. Such visualizations lend a degree of interpretability to the model's decision process.

4.2 Role of Hyperparameters

Importance of Temperature: Fig. 4.6 presents the performance by varying $\Delta_\tau = \tau_{eval} - \tau_{train}$, the difference in temperature used in evaluation versus train time on the Dermnet dataset. We can see that while performance is agnostic for base classes, higher interpolation through an increased temperature leading to $\Delta_\tau > 0$ leads to improved performance on novel classes. Conversely, when $\Delta_\tau < 0$, performance drops. This suggests that at evaluation time, the model requires interpolating between the cluster prototypes to effectively predict a class label.

Does episodic memory help? Table 4.5 shows that we can get improvements even with a simple online update rule that blends prototypes computed using the support set in the current episode with the past, using $\alpha = 0.5$. This trend is also seen for prototypical networks (denoted by PN*). We leave as future work the task of modeling adaptive α.

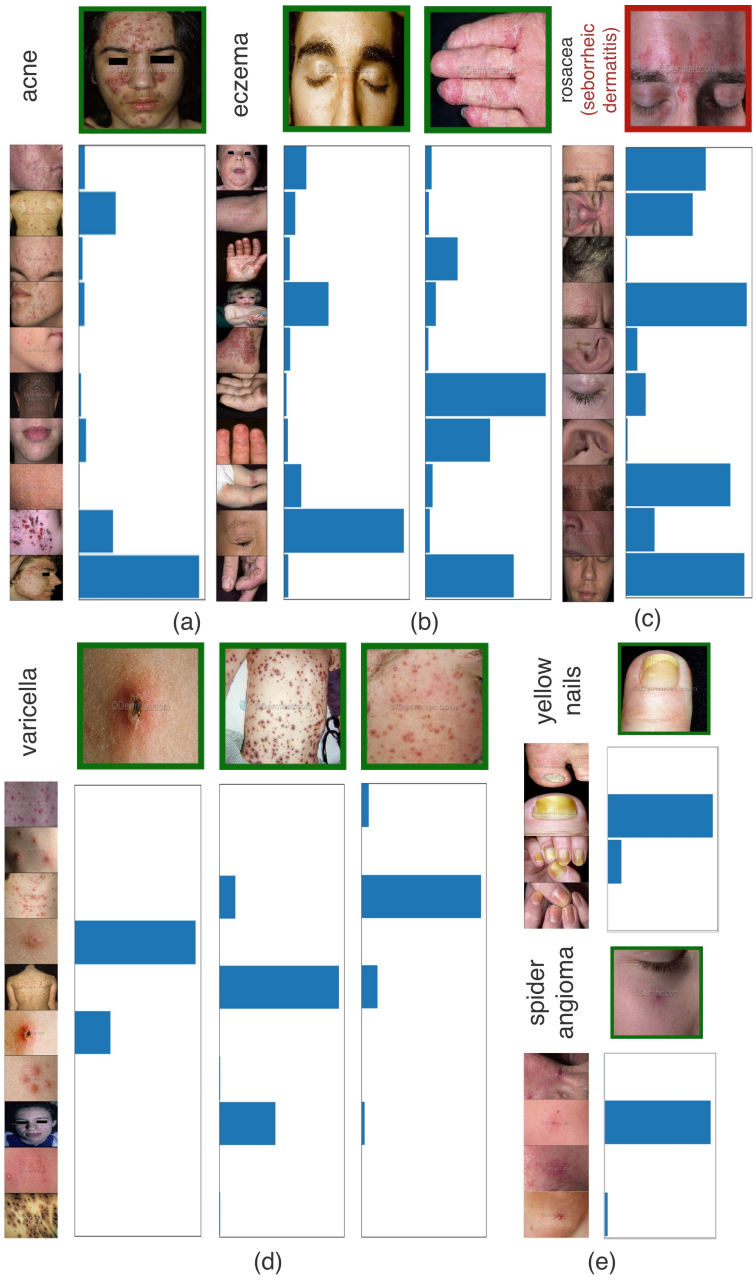

Figure 4.4: Effectiveness of using multiple clusters. Shown for base and novel classes. (a), (b), (d), (e): Examples from test set that are correctly classified by PCN. For each class, we show the nearest neighbor to the learned prototypes. We also present examples (columns) whose labels are correctly predicted and the inferred cluster responsibilities $q(z|c, x)$ conditioned on the correct class. (c): Test set example that is misclassified by PCN. Correct label is shown in black, while the incorrect prediction is shown in red. We show the nearest neighbors to the learned cluster prototypes of the *predicted* (incorrect) class, and the corresponding cluster responsibilities. Note that green outlines around query images denote correct classification while red denotes incorrect classification. (e) corresponds to novel classes.

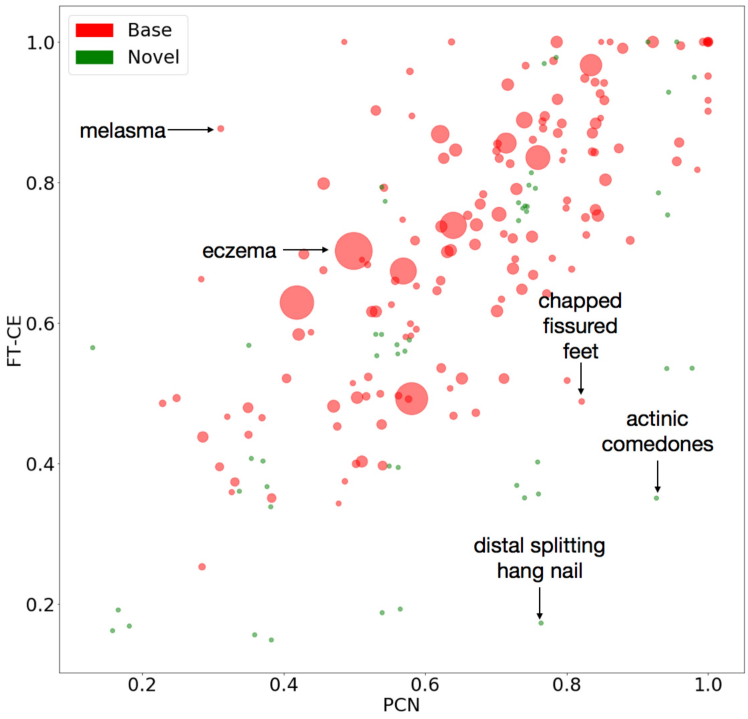

Figure 4.5: Comparison between FT_{200}-CE and PCN: Per-class accuracy. Each class is denoted by a dot and the area of dot is proportional to the number of training examples for the class.

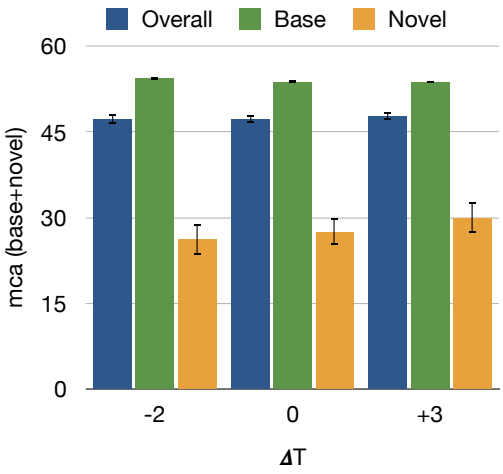

Figure 4.6: Effect of temperature on PCN

Table 4.5: Importance of episodic memory

Approach	α	$\text{mca}_{\text{base+novel}}$
PCN	0	45.62 ± 0.89
PCN	0.5	47.49 ± 0.71
PN	0	44.35 ± 0.53
PN*	0.5	45.84 ± 0.46

Table 4.6: Recall@k on top 200 classes.

	Approach	r@5$_{\text{base+novel}}$	r@5$_{\text{base}}$	r@5$_{\text{novel}}$	r@10$_{\text{base+novel}}$	r@10$_{\text{base}}$	r@10$_{\text{novel}}$
n=5	FT_{200}-CE	**77.7 ± 0.8**	**80.8 ± 0.8**	38.0 ± 2.1	84.9 ± 0.5	**88.1 ± 0.4**	45.3 ± 1.5
	PN	70.9 ± 0.4	71.5 ± 0.3	**63.2 ± 2.3**	80.2 ± 0.3	80.5 ± 0.2	**76.3 ± 2.1**
	PCN (ours)	**77.7 ± 0.2**	**79.2 ± 0.2**	**59.2 ± 2.6**	**86.0 ± 0.4**	87.1 ± 0.2	**72.2 ± 3.6**
n=10	FT_{200}-CE	**78.6 ± 0.1**	**80.2 ± 0.3**	58.0 ± 3.2	**86.4 ± 0.3**	**88.1 ± 0.1**	65.9 ± 2.6
	PN	69.4 ± 0.3	69.1 ± 0.3	**72.2 ± 1.7**	78.6 ± 0.3	78.1 ± 0.3	**84.7 ± 2.2**
	PCN (ours)	76.3 ± 0.2	76.4 ± 0.2	**74.5 ± 2.6**	85.0 ± 0.2	85.0 ± 0.3	**85.0 ± 2.0**

4.3 Per-class Accuracy

In Figure 4.5 we provide a class-wise performance comparison on the Dermnet dataset between the PCN and FT_{200}-CE models as a scatter plot (shown here for the best performing PCN model evaluated with a train shot of 10 with $\text{mca}_{\text{base+novel}} = 50.92$).

We make the following observations: Overall metrics indicate that FT_{200}-CE demonstrates slightly stronger average performance on base classes. For a large fraction of the base classes, both methods have similar performance. For the ones in which PCN performance is lower, there is usually a reasonable lower bound on the classification accuracy. In contrast, for novel classes, PCN performs better on average. Importantly, when FT_{200}-CE performance is lower than PCN, it is usually significantly lower. As an example is the novel class 'distal splitting hang nail' for which PCN performs significantly better.

4.4 Additional Metrics

In Table 4.6 we provide recall@5 and recall@10 metrics for PN, PCN, and FT_{200}-CE approaches, for train shot $n = 5$ and $n = 10$. PCN performs on par with the FT_{200}-CE baseline and outperforms PN on these metrics. We note that since our test set is imbalanced, recall@k metrics unfairly reward strong performance on the head classes (which is observed with FT_{200}-CE). However, it is clear that PCN and PN models dominate in recall@k metrics on novel classes.

4.5 Applicability to *mini*ImageNet

While we develop our approach in the specific context of dermatological diagnosis, we investigate the applicability of our model to other datasets. We focus on the *mini*ImageNet dataset [27], which is a standard benchmark in the few-shot literature. While an order of magnitude smaller than the original ImageNet dataset, the dataset exhibits moderate intra-class diversity which makes it a well-suited testbed for our approach.

4.5.1 Results on *mini*ImageNet

As a sanity check, we evaluate performance of our approach on the *mini*ImageNet dataset [27]. We use the splits proposed in [39], and match our implementation and training to prior work [1, 27]. Our embedding network is a CNN architecture comprised of four convolutional blocks, that leads to 1600-dimensional embeddings.

Table 4.7: *mini*ImageNet accuracy on $K_{base_2} \cup K_{novel-test}$

	n=1		n=5	
	5-way	20-way	5-way	20-way
PN	56.58	44.43	70.83	48.29
PCN	56.70	44.97	74.93	53.49

Table 4.8: Accuracy on *mini*ImageNet. We report the mean of the accuracies with 95% confidence intervals. Note that PFA-Simple corresponds to the 'Ours-Simple' model reported in [32].

	5-shot 5-way
Baseline-NN [39]	51.0 ± 0.7%
MN [27]	55.3 ± 0.7%
MAML [45]	63.1 ± 0.9%
PFA-Simple [32]	**67.9 ± 0.2%**
PN [1]	65.3 ± 0.7%
PCN (ours)	**66.9 ± 0.7%**

Recall that our primary objective is learning *without forgetting*. To evaluate this, we modify the existing *mini*ImageNet benchmark as follows: Similar to [33, 29], we train a base

classifier and validate it on heldout data from base classes K_{base} [2](representation learning phase). Next, in the low-shot learning phase, we split K_{base} into disjoint subsets $K_{base_1} \cup K_{base_2}$. We cross-validate hyperparameters on $K_{base_1} \cup K_{novel-val}$ classes, and report averaged accuracy over test 600 episodes on $K_{base_2} \cup K_{novel-test}$ classes. Note that as before, both PN and PCN get access to the appropriate subset of the training set to generate prototypes for base classes.

Table 4.7 presents results of various classification settings, with a test-query size of 15 and averaged over 600 test episodes. As seen, PCN strongly outperforms PN in the 5-shot case and has similar performance in the 1-shot case, where interpolation for novel classes is not possible. Our best performing PCN model is trained with 5 clusters per class, and uses 1 cluster for novel classes and a temperature value of 4.0 (all values picked via grid search).

We further report numbers on the standard *mini*ImageNet benchmark in Table 4.8, and compare against recently proposed work. We note that while several other approaches have also been proposed that report performance using deeper residual features, for fairness we only compare against approaches that use identical CNN architectures. We observe that our model outperforms several prior approaches and is competitive with the current state of the art. As with prior work, we train our model by monitoring generalization performance on unseen novel classes from the validation set, and report performance on 5-shot-5-way performance averaged over 600 episodes from the test split. We train with a way of 20 at train time. We use the identical CNN architecture as in [1], with 4 convolutional blocks, leading to 1600-dimensional embeddings for the 84x84 images in the *mini*ImageNet dataset. Each block is structured as follows: a 64-filter 3x3 convolution layer, batch normalization layer, ReLU nonlinearity, and 2x2 max-pooling layer. We train our models via SGD with Adam [43], starting with a learning rate of 1e-3 that we anneal by half every 2000 episodes. Note that we our PCN model is built on top of our PN reimplementation, and for fair comparison we report performance with our implementation of Prototypical Networks in row 3, which achieves slightly lower performance than that reported in [1].

[2]As in [33], we sample an additional 300 examples per base class for each of validation and test

CHAPTER 5
CONCLUSION AND FUTURE WORK

In this thesis, we propose Prototypical Clustering Networks: a few shot learning approach to dermatological image classification. This method is scalable to novel classes, and can effectively capture intra-class variability. We observe that our approach outperforms strong baselines on this task, especially on the long tail of the data distribution. Such a machine learning system can be a valuable aid to medical providers, and improve access, equity, quality, and cost-effectiveness of healthcare.

There exist a number of future directions worth pursuing. The true effectiveness and utility of our system is in aiding the physician, and this requires studies that include such a deployment. Immediate extensions include determining the absence of any condition (adding *i.e.* a 'normal' class), and controlling for demographic variables when appropriate. Another interesting direction would be to incorporate additional modalities of data for more robust prediction. While our approach learns features end to end, another promising avenue might be fusing such learned representations with features designed using domain expertise, such as diagnostic markers typically tracked by dermatologists. Dermatologists use symptoms that patients experience, such as itchiness of the skin, in disambiguating skin conditions [14]. Incorporating these medical symptoms as part of the classification task will be an interesting direction to pursue.

REFERENCES

[1] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[2] WHO, "Telemedicine: Opportunities and developments in member states: Report on the second global survey on ehealth," 2009.

[3] Strehle and Shabde, "One hundred years of telemedicine: Does this new technology have a place in paediatrics?" *Archives of Disease in Childhood*, pp. 956–959, 2006.

[4] A. Dasgupta and S. Deb, "Telemedicine: A new horizon in public health in india," *Indian journal of community medicine : official publication of Indian Association of Preventive and Social Medicine*, vol. 33, 2008.

[5] C. O. Bagayoko, D. Traore, L. Thevoz, S. Diabate, D. Pecoul, M. Niang, G. Bediang, S. T. Traore, A. Anne, and Geissbuhler, "Medical and economic benefits of telehealth in low- and middle-income countries: Results of a study in four district hospitals in mali," *BMC health services research*, vol. 14, 2014.

[6] O. e. a. Faye, "A teledermatology pilot programme for the management of skin diseases in primary health care centres: Experiences from a resource-limited country (mali, west africa)," *Tropical medicine and infectious disease*, vol. 3, 2018.

[7] NHANES, "Skin conditions and related need for medical care among persons 174 years," *NHANES: United State*, 1978.

[8] D. Bickers, H. Lim, and D. Margolis, "The burden of skin diseases: 2004 a joint project of the american academy of dermatology association and the society for investigative dermatology," *Journal of American Academy Dermatology*, 2006.

[9] J.K.Scholfield, D. Grindlay, and H. Williams, "Skin conditions in the uk: A health needs assessment," *University of Nottingham, Centre of Evidence Based Dermatology UK; Nottingham, UK*, 2009.

[10] R. Hay and L. Fuller, "The assessment of dermatological needs in resource-poor regions," *International Journal of Dermatology*, 2012.

[11] M. Basra and M. Shahrukh, "Burden of skin diseases," *Expert Review Pharmacoeconomics Outcomes Research*, 2009.

[12] D. Federman, J.Concato, and R. Kirsner, "Comparison of dermatologic diagnoses by primary care practitioners and dermatologists. a review of the literature," *Archives of Family Medicine*, vol. 8, no. 2, 1999.

[13] E. E. Goldman, "Skin diseases get misdiagnosed in primary care," *Family Practice News*, 2007.

[14] J. Resneck, M. Abrouk, M. Steuer, and et al, "Choice, transparency, coordination, and quality among direct-to-consumer telemedicine websites and apps treating skin disease," *JAMA Dermatology*, vol. 152, no. 7, 2016.

[15] "World health organization, international statistical classification of diseases and related health problems: Chapter xii: Diseases of the skin and subcutaneous tissue,"

[16] E. Wilmer, C. Gustafson, C. Ahn, S. Davis, S. Feldman, and W. Huang, "Most common dermatologic conditions encountered by dermatologists and nondermatologists," *Cutis*, vol. 94, no. 6,

[17] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, p. 115, 2017.

[18] Z. Li, C. Wang, M. Han, Y. Xue, W. Wei, L. Li, and F. Li, "Thoracic disease identification and localization with limited supervision," *IEEE Computer Vision and Pattern Recognition*, 2018.

[19] H. Liao, "A deep learning approach to universal skin disease classification," *University of Rochester Department of Computer Science, CSC*, 2016.

[20] X. Sun, J. Yang, M. Sun, and K. Wang, "A benchmark for automatic visual classification of clinical skin disease images," in *European Conference on Computer Vision*, Springer, 2016, pp. 206–222.

[21] J. Yang, X. Sun, J. Liang, and P. L. Rosin, "Clinical skin lesion diagnosis using representations inspired by dermatologist criteria," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1258–1266.

[22] G. Van Horn and P. Perona, "The devil is in the tails: Fine-grained classification in the wild," *arXiv preprint arXiv:1709.01450*, 2017.

[23] Y.-X. Wang, D. Ramanan, and M. Hebert, "Learning to model the tail," in *Advances in Neural Information Processing Systems*, 2017, pp. 7029–7039.

[24] X. Zhu, D. Anguelov, and D. Ramanan, "Capturing long-tail distributions of object subcategories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 915–922.

[25] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[26] V. Fragoso and D. Ramanan, "Bayesian embeddings for long-tailed datasets," 2018.

[27] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 3630–3638.

[28] E. Triantafillou, H. Larochelle, J. Snell, J. Tenenbaum, K. J. Swersky, M. Ren, R. Zemel, and S. Ravi, "Meta-learning for semi-supervised few-shot classification," 2018.

[29] B. Hariharan and R. B. Girshick, "Low-shot visual recognition by shrinking and hallucinating features.," in *ICCV*, 2017, pp. 3037–3046.

[30] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," *arXiv preprint arXiv:1605.06065*, 2016.

[31] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5822–5830.

[32] S. Qiao, C. Liu, W. Shen, and A. Yuille, "Few-shot image recognition by predicting parameters from activations," *arXiv preprint arXiv:1706.03466*, vol. 2, 2017.

[33] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4367–4375.

[34] E. Schönfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero-and few-shot learning via aligned variational autoencoders," *arXiv preprint arXiv:1812.01784*, 2018.

[35] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "Icarl: Incremental classifier and representation learning," in *Proc. CVPR*, 2017.

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[37] S. Fort, "Gaussian prototypical networks for few-shot learning on omniglot," *arXiv preprint arXiv:1708.02735*, 2017.

[38] R. Boney and A. Ilin, "Semi-supervised and active few-shot learning with prototypical networks," *arXiv preprint arXiv:1711.10856*, 2017.

[39] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 2016.

[40] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, p. 2007, 2007.

[41] C. Zauner, "Implementation and benchmarking of perceptual image hash functions," 2010.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[44] S. R. Fuller, G. M. Bowen, B. Tanner, S. R. Florell, and D. Grossman, "Digital dermoscopic monitoring of atypical nevi in patients at risk for melanoma," *Dermatologic Surgery*, vol. 33, no. 10, pp. 1198–1206, 2007.

[45] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400*, 2017.