SUPER PARETOS: BAYESIAN ACTIVE META LEARNING FOR SPATIAL TRANSFERABILITY OF DEEP LEARNING MODELS

SRIKUMAR SASTRY
June, 2022

SUPERVISORS:

dr. M. Belgiu
dr. R. Vargas Maretto

SUPER PARETOS: BAYESIAN ACTIVE META LEARNING FOR SPATIAL TRANSFERABILITY OF DEEP LEARNING MODELS

SRIKUMAR SASTRY
Enschede, The Netherlands, June, 2022

Thesis submitted to the Faculty of Geo-information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: M-GEO

SUPERVISORS:

dr. M. Belgiu
dr. R. Vargas Maretto

THESIS ASSESSMENT BOARD:

prof.dr.ir. A. Stein (chair)
dr. C. Paris, Department of Natural Resources (External examiner)

ABSTRACT

A common objective to address in the current realm of artificial intelligence (AI) is achieving more with less. This follows the notion of pareto which attributes 80% of results to just 20% of inputs. With advancing technology and research, it has become possible for AI to achieve super pareto, which further widens the ratio between necessary inputs and corresponding expected results. Keeping this in mind, we can draw direct correspondences between paretos and data efficiency. Data inadequacy has been a common problem in modern deep learning applications. It becomes a serious challenge especially in remote sensing applications where data collection and annotation is time consuming and expensive. A plethora of methods have been proposed in the past that fall into some theme of semi-supervised learning, active learning or transfer learning. These methods promise to solve the problem of data inadequacy and uphold the super pareto notion. However, a simple survey of these methods shall highlight their limited practical applicability on real life datasets. Such methods not only are computationally expensive but also fail to perform in varying realistic settings. In this study, we focus on one of the themes, namely active learning, commonly used to address the issue of efficient data assimilation. We take a holistic approach and describe its possible applications in imaging science. Most studies fail to uphold the chief principle that active learning was built on. We elaborate these limitations further in this study and propose novel active learning methods for image classification and segmentation. We not only compare our method with existing baselines but also present their results on real life remotely sensed datasets. One of the applications we focus on is crop mapping. Automatic crop mapping is important to ensure food security and efficient crop management. With a traditional deep learning approach, we are able to achieve an accuracy of 79.34%. With active learning, we are able to achieve a similar result but with only 1.31% of the data previously used. This translates to reducing about 75x efforts required for data collection and annotation. We see that this gap between necessary inputs and results is especially noticeable in practical datasets and therefore believe the notion of super pareto to become the new normal. Further in the study, we present the online learning setup for few shot learning which is extremely common in the real world. To this end, we propose an active meta learning method to understand the advantages of meta learning over offline/batch learning.

Keywords

Active Learning, Uncertainty, Meta Learning, Bayesian Inference, Few Shot Learning

ACKNOWLEDGEMENTS

I believe that the contributions and outcomes of this work would not have been possible without the constant teaching, guidance and encouragement of all my teachers. Even after the end of my affiliation with the institute, they shall continue to be a source of inspiration and motivation.

I would like to express immense gratitude towards my supervisors Dr. Mariana Belgiu and Dr. Raian Vargas Maretto for their endless support and input over the coarse of my MSc research. They witnessed my achievements as well as my setbacks and walked me through thick and thin. During the entirety of the research, I couldn't wait to present every minor result of my implementations and see a smile on their faces. Both of them set a high standard for research which motivated me to deliver at my best. They were patient mentors, listening to my bad ideas, allowing me to implement them, watching me fail and finally making me learn from them. Dr. Belgiu has always taught me to look for the positives in every situation. She has always insisted me to reflect on results critically. Nevertheless, she believes in constantly adapting and moving on. Dr. Maretto has always brought interesting ideas to the table. His advice and mentorship is like water in a desert, impossible to find everywhere.

I would like to extend my appreciation towards my thesis assessment board - Dr. Alfred Stein and Dr. Claudio Persello. They helped me connect the theoretical aspects of my work with the reality. Further, their inputs helped me streamline the scope and objectives of my research according to the stipulated timeline. I thank Dr. Nathan Jacobs, who accepted the request to be my internship supervisor. He provided several exciting perspectives of my work that added depth to the overall MSc research. I would like to thank the hard working team - Sina Mohammadi, Qi Dong and Chenxi Duan. The insightful discussions and their interesting questions helped me identify the limitations of my work. I wish them all the best for their future doctoral research. I want to extend my credits to Dr. Wan Bakx for providing valuable inputs towards my MSc proposal and being there as an advisor for the thesis assessment board.

This work would have been incomplete without the availability of powerful computing resources. To this end, I thank the EOS department for giving me access to their cloud computing servers. Dr. Frank Osei and Ali Bagislayici for entertaining my requests and solving all the issues related to this server. I thank the CRIB department for providing the crib geospatial computing hub. It really helped me compile my results. Dr Serkan Grigin for supporting me solve issues related to this platform. It is also not late to acknowledge the developers of QGIS, Python and R for making them freely available. Most of my work has been accomplished using these tools.

Finally, I would like to acknowledge my family, and friends at ITC for believing in me. They were a constant source of motivation and critical feedbacks. They helped me settle in the Netherlands and made sure that I had a good time every single day.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

> One of the very worst uses of time is to do something very well that need not be done at all.
>
> Brian Tracy in *Eat That Frog!*, 2001

1.1 BACKGROUND

The complexity of deep learning models is increasing with every new iteration. The number of parameters in recently proposed models are reaching unimaginable levels. However, the success of these complex models are only limited to setups with high computational processing capability and availability of enough annotated data. Such setups are rarely accessible and difficult to construct in the first place. The two research questions that are evident from this observation are -

"How can good results be achieved with a simple model but with abundant annotated data?"
"How can good results be achieved with an arbitrary model with scarce annotated data?"

To the best of our knowledge, only one technique - knowledge distillation (Hinton et al., 2015) tries to address the former question. However, it requires pretrained deep networks for knowledge transfer to smaller networks. Futher, we believe that the theory of statistical learning shall limit the generalization of smaller networks on unseen data. In this thesis, we focus on the latter question which is a relevant problem for the current machine learning and remote sensing community.

Image classification and segmentation tasks are highly relevant in computer vision and remote sensing applications. For these problems, the most common methods used for generating labelled samples include: 1) in-situ measurements; 2) using already referenced maps; 3) imagery interpretation (Bruzzone & Persello, 2009). The cost of such methods is usually very high. Therefore, only a few training pixels or images are available for majority of image classification tasks. Limited availability of annotated samples not only affects the performance of classification at hand but also limits the model's spatial and temporal generalizability. Ball et al., (2017) presented nine shortcomings of deep learning in remote sensing which also includes the problem of unavailability of data.

In recent context, limited label classification task is known as semi-supervised learning. In semi-supervised learning, the classifier network has access to a few labeled data points and a large

pool of unlabeled data points. Using approaches such as unsupervised learning and contrastive learning, the network may benefit from the pool of unlabeled data during training. Few Shot Learning (FSL) is a special subset of semi-supervised learning, where the labeled pool of data has a fixed structure (Y. Wang et al., 2020). FSL problems are commonly characterized as N-way k-shot classification problems, where N is the number of classes and k is the number of labeled samples available per class for training at any instance. When k=0, FSL is known as zero-shot learning and when k=1, it is referred as one-shot learning. Four main techniques are used to tackle a FSL task. These techniques are briefly explained below.

1. Generative Modelling (Harshvardhan et al., 2020)
2. Transfer Learning (Zhuang et al., 2021)
3. Active Learning (Settles, 2010)
4. Meta Learning (Hospedales et al., 2021)

Generative models can produce synthetic samples which statistically mimic the original data. Such models can increase the size of the training set, thus improving the generalizability of the discriminator model. Several generative models proposed in the last decade have achieved state-of-the-art performance (Dhariwal & Nichol, 2021; Goodfellow et al., 2014; Kingma & Welling, 2014). In remote sensing, several studies (Davari et al., 2019) have used gaussian mixture model (GMM) based synthetic sample generators for data augmentation. He et al., (2017) used spectral-spatial features to train a generative adversarial network (GAN). Although these methods are powerful, they generate distributions that are not an exact representation of the original distribution. Shorten and Khoshgoftaar, (2019) noted that data augmentation methods are more suited for reducing overfitting and performing oversampling.

Transfer learning is used to improve the performance of a learner in one domain using information from some other related domain. In remote sensing, transfer learning can greatly reduce the amount of training samples required in the target domain. Xie et al., (2016) has used chained transfer learning graphs with CNN to map poverty levels. Zhang et al., (2019) uses cross-sensor strategy which can adapt to different source and target sensors. Up until now these methods have focused only on fine tuning parameters and domain adaptations. Multitask learning is a type of transfer learning which uses multiple datasets at the same time to cotrain a single network. Liu and Shi, (2020) reduces overfitting in multitask learning context by using single feature vector from multiple datasets. Bischke et al., (2019) uses cascaded multitask loss to incorporate geometry of objects during segmentation. The drawback of multitask learning is the assumption that multiple datasets share common features.

Active learning reduces the cost of labelling samples by selecting the most useful unlabeled samples. These selected samples are usually annotated by a human oracle. There have been various active learning strategies proposed in the literature including query by committee, least confidence, minimum margin, maximum entropy, diversity sampling and so on (Settles, 2010). Gal et al., ((Gal et al., 2017)) was the first to introduce active learning with DL. In remote sensing, active learning methods have mostly focused on using SVM (Demir et al., 2011; Pasolli et al., 2011). Débonnaire et al., (2016) has used temporal clustering as input to sample selection algorithm with Random Forest and SVM for classification. Only recently, deep active learning has been introduced in remote sensing (Qu et al., 2020; Rodríguez et al., 2021; Růžička et al., 2020). However, the practical

applicability of present active learning approaches (in remote sensing) is still an open question as online annotation is not always possible in remote sensing applications involving ground surveys.

Meta learning tries to use prior (meta) information of training on related tasks to improve predictions on unseen tasks. Unlike multitask learning, meta learning only tries to improve performance on unseen tasks. Siamese networks (van der Spoel et al., 2015), matching networks (Vinyals et al., 2016), prototypical networks (Snell et al., 2017), relation networks (Sung et al., 2018) and their variants are the state-of-the-art deep learning architectures in FSL. Finn et al., (2017) and Nichol et al., (2018) describe optimization based meta learners. In remote sensing, meta learning is relatively new and only a few studies have used it. Liu et al., (2019) used siamese network with metric learning regularization term while Tang et al., (2020) used prototypical network with spectral-spatial feature extraction algorithm for aerial scene classification. Ruswurm et al., (2020) used Model Agnostic Meta Learning (MAML) for few shot land cover classification and compared with parameter based transfer learning.

1.2 RESEARCH GAPS

We raise some fundamental questions in this thesis that are seldom debated open or that form basis of novel research. The main theme of focus in the thesis is optimality and feasibility of active learning methods. Though active learning appears powerful on paper, its practical applicability depends on the setting we are working with. This includes the process of annotation in use, type of computational resources available and so on. Most active learning methods proposed in the past have been validated only on toy and/or synthetic datasets. Also, they perform well only with certain training settings (Munjal et al., 2020). This raises an important question on their feasibility on large scale and real life datasets. In this thesis, we present results of active learning methods on practical remote sensing datasets.

In the literature, there is no clear discussion whether active learning methods developed for image classification may be successfully applied for semantic segmentation (and vice versa). We believe that the definition of active learning changes when we work with distinct computer vision tasks. Do we experiment on the image level, pixel level or object level? There is no clear answer on how to compare active learning methods that function on different image levels. To this end, we try to provide the idea for adapting active learning methods between image classification and semantic segmentation tasks.

Online active learning setup using deep learning has not been investigated much in the literature. Online learning processes only the new incoming data in each training iteration. FSL can also be considered within online learning setups. However, it is also limited on experiments in the literature. Further, there is no formal definition of FSL for semantic segmentation. In practical applications, we may not have access to a large number of tasks for the FSL to be successful. Where do the tasks come from in a real world application? Why should the network be provided data points in a few shot fashion? How does FSL compare with a vanilla batch learning approach? All these observations question the rationality of FSL. In this thesis, we try to formulate a framework of FSL for semantic segmentation. Specifically, we address the problem of crop classification. We use MAML for few shot crop mapping and compare it with batch learning approach. Active learning is used in this case to query labels.

1.3 RESEARCH OBJECTIVES AND QUESTIONS

Overall Objective. The main objective of this thesis is to combine active learning with a meta learning framework to address the problem of data inadequacy and improve the generalization of deep learning models across space.

Sub Objective 1. Design a novel query strategy for active learning for image classification.

Sub Objective 2. Evaluate the performance of proposed active learning method against state-of-the-art methods.

Sub Objective 3. Evaluate the performance of proposed active learning method on semantic segmentation task.

Sub Objective 4. Design a novel query strategy for active learning for semantic segmentation.

Research Question 1. How to incorporate uncertainty and diversity in batch mode active learning?

Research Question 2. What is the performance of active learning for image classification applied on semantic segmentation task?

Research Question 3. How to incorporate class diversity in active learning for semantic segmentation?

Research Question 4. How to improve spatial transferability of active learning strategy?

Sub Objective 5. Combine active learning with meta learning for improving spatial transferability.

Research Question 5. Does the combination of active learning and meta learning improve spatial transferability?

1.4 CONCEPTUAL FRAMEWORK

The figure 1.1 describes the building blocks used in the entire thesis. We begin with raw unlabeled remotely sensed data. This data is fed into an active learning loop that is used to obtain label for some of these unlabeled data points. Active learning consists of an acquisition function that is used to query informative and representative unlabeled data points. These queried data points are labeled by a human expert (oracle). However, in the entirety of the thesis, we use already existing reference datasets to obtain labels for the queried data points. The labeled dataset is passed to the meta learning framework. The dataset is divided into a support set and a query set. The support set is used to compute task specific parameters and the query set is used to update the classifier with average set of parameters of the tasks. A single task is a set of support and query set. A set of tasks form an episode that is used in a single training iteration of meta learning. There may be different forms of meta learning, but we focus on optimization based meta learning.

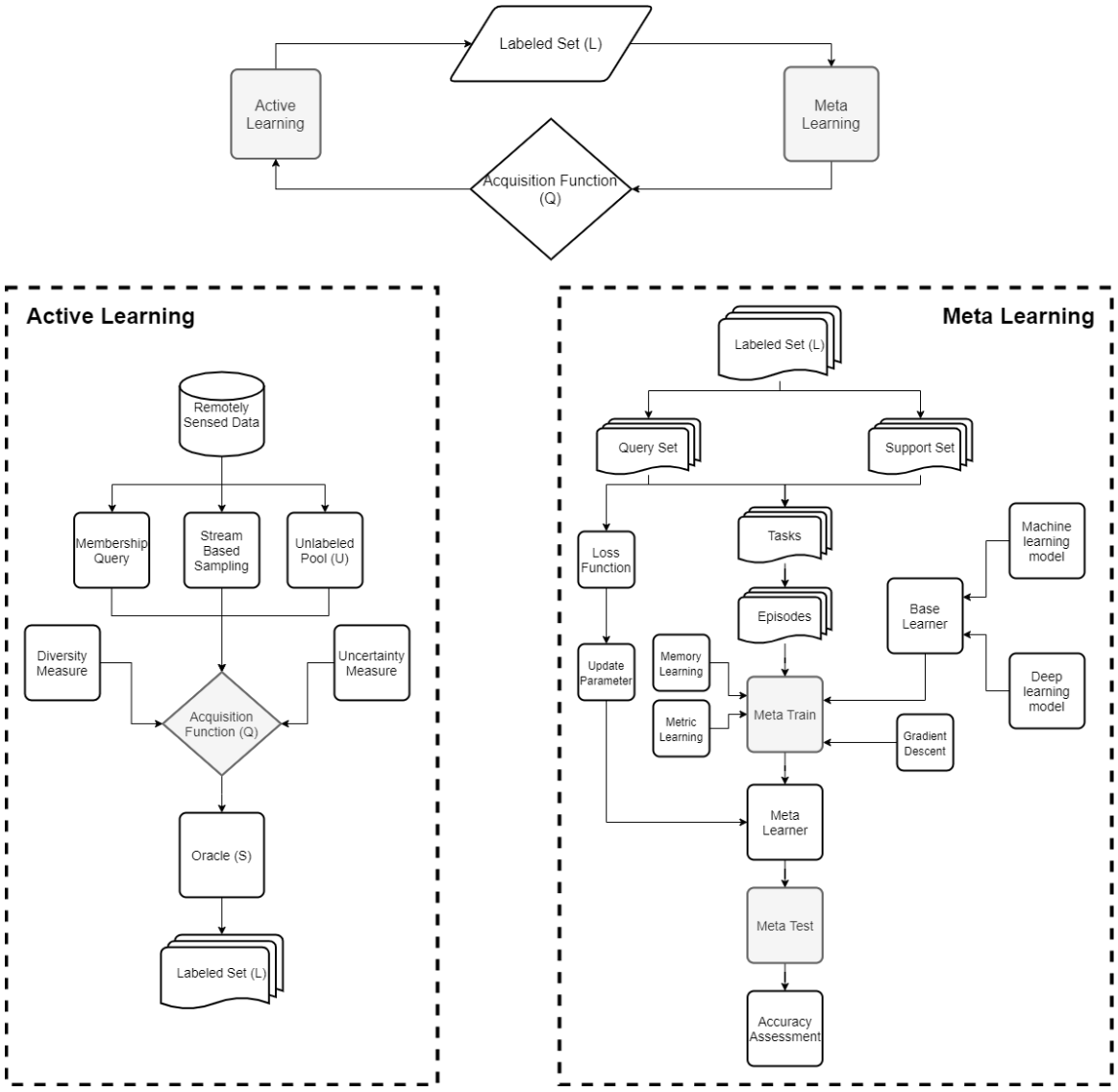


Figure 1.1: Conceptual diagram depicting the data flow between the active learning and meta learning framework.

1.5 THESIS ORGANIZATION

This thesis is organized in four chapters which are in a chronological sequence. We begin by developing an active learning method for image classification. We also formally define the term active learning in this section. In the next section, we discuss the limitations of active learning methods (used for image classification) applied on semantic segmentation tasks. We propose a novel active segmentation algorithm for per-pixel classification of images. We present the description of the study area and dataset used in this section. The third section deals with the problem of few shot learning of models which we address using active meta learning. The final section discusses some limitations of the proposed work and presents some recommendations for future work. Here we present a brief summary of the first three chapters -

Chapter 2: In this chapter, we propose a simple yet highly efficient and robust active learning (AL) framework for image classification. Most of the existing AL strategies are either not scalable with increasing acquisition batch sizes or not robust to noise. They select samples greedily without considering the acquisition state of previous iteration. Further, very little focus has been given to the selection of the initial seed set for active learning. In this work, we propose a new framework that combines simulated annealing within AL to select those samples which improve their acquisition cost in the previous iteration. A convex combination of a diversity measure and an uncertainty measure is used as the acquisition cost. The diversity measure ensures consistent prediction of samples lying farthest from the decision boundaries and, eventually, an unbiased estimation of uncertainty. We demonstrate the efficiency and robustness of our proposed framework over the current state-of-the-art AL strategies using Bayesian CNNs.

Chapter 3: In this chapter, we propose an active learning method that is highly effective for spatial transferability in semantic segmentation tasks. Most of the current active learning methods for semantic segmentation do not consider heterogeneity of patches for their acquisitions. Further, they use average statistics of pixels as the acquisition cost of a patch which might be highly biased for complex patches. Also, very little focus has been given on active learning for spatial transferability. In this work, we combine class distribution with their predictive uncertainties that is conditioned on the target site. In this way, we only select patches in the training area that are relevant for performance improvement in the testing area. We demonstrate the success of our proposed method on the task of crop classification in the United States using Sentinel-2 imagery. Further, we compare our method with those active learning methods which use average statistics of pixels.

Chapter 4: In this chapter, we present an online learning setup for few shot learning. Most of the studies have ignored sequential nature of data in case of multi task or few shot learning. We address this challenge by proposing a framework that combines the active learning method previously proposed with meta learning. Meta learning, built on the principle of learning to learn, has been successfully applied for knowledge transfer, reinforcement learning and unsupervised learning. However, meta learning is easily prone to overfitting in scarce data experiments. Further, the definition of meta learning for few shot segmentation hasn't been clearly described in previous works. In this work, we formally define few shot learning for semantic segmentation. We present preliminary results of the proposed active meta learning method for few shot crop classification. Further, the results are compared with previously described batch mode active learning and random sampling methods.

Chapter 2

Active Learning

> Before you begin scrambling up the ladder of success, make sure that it is leaning against the right building.
>
> Steven Covey in *The 7 Habits of Highly Effective People*, 1989

2.1 INTRODUCTION

Deep learning has evolved as a promising approach to solve supervised classification tasks. However, the success of this approach is limited only to data abundant tasks. In real world, collecting and labeling a large dataset for classification is expensive and time consuming. Active learning is a technique that can be used to achieve data efficiency in sparse label problems (Cohn et al., 1996). Active learning works by building the training set iteratively by querying the most informative unlabeled samples. The informativeness is measured by an *acquisition function* and the acquired samples are labeled by an *oracle*.

The most common acquisition functions use uncertainty as a measure of informativeness. In deep active learning, bayesian neural networks (BNN) are popularly used for modeling uncertainty and have been proven to be very effective (Vineeth & Jain, 2021). However, active learning for deep learning faces the problem of high dimensionality of the network. Second, uncertainty is evaluated using the network's prediction itself which might be highly biased. The first problem is addressed by using approximate inference techniques of which Monte Carlo (MC) dropout (see A) is a widely used technique (Gal & Ghahramani, 2015, 2016).

The second problem exists due to the initial seed set problem (Yang et al., 2015). Active learning algorithms usually start with a randomly selected subset of samples from the unlabeled pool. These samples are used to pretrain the network for subsequent uncertainty evaluations. It is assumed that after pretraining, the network is confident on samples farthest from the decision boundary. However, the randomly selected samples might be of poor quality and lead to biased uncertainty evaluations. The initial performance of an active learning algorithm can highly affect its overall performance in the long run (Chandra et al., 2020).

One of the key challenges to active learning algorithms in recent times is the ability to scale well with the size of unlabeled pool. This challenge has been addressed previously by acquiring

batch of samples and training the model less frequently. Most commonly, the top K informative samples are selected in each iteration. However, this may lead to mode collapse where samples are informative but not necessarily dissimilar (Pop & Fulop, 2018).

There have been density based active learning method proposed in the past (Ash et al., 2019; Sener & Savarese, 2018; Zhdanov, 2019) which select informative samples that are as diverse as possible. However, majority of these methods perform clustering at each iteration of acquisition. The run time of such methods increase exponentially with increase in the acquisition batch size (Citovsky et al., 2021). Recently, diverse batch mode active learning methods (Kirsch et al., 2019; Woo, 2021) have been developed. But, none of them scale well with increasing acquisition batch size. On top of it, most of these methods are not robust to varying dataset (noise, class imbalance) or training settings (Munjal et al., 2020).

Another technique to make active learning algorithms scalable is to process a random subset of unlabeled pool (active pool) in each iteration instead of processing the entire unlabeled pool. However, if the size of this active pool is too small, it may under represent the entire pool (especially in large imbalanced datasets). Furthermore, the active pool may contain similar or poor quality samples due to randomness, which might be a trade off to performance. Here, we address these challenges by selecting those samples which improve the overall acquisition cost in the previous iteration and generating the active pool from a stratified unlabeled pool.

In this work, we propose to use simulated annealing (see B) for probabilistic selection of samples. This allows for a **less** greedy selection of samples. For the acquisition cost, clustering and distance of sample to corresponding cluster center is used as the diversity measure. The key advantage of this framework is that clustering is performed only once. Uncertainty is modeled using MC dropout and quantified using the margin score (Scheffer et al., 2001). These two measures are normalised and their convex combination is used as the overall acquisition cost. The contribution of this work is fourfold:

- The proposed method solves the initial seed set problem by selecting the most diverse samples in the beginning. It also prevents mode collapse by sampling the active pool from clusters.
- The proposed method is less greedy than existing active learning strategies. It tries to improve the acquisition cost of each cluster independently. Furthermore, the probabilistic selection feature of simulated annealing allows the algorithm to escape maxima of acquisition costs so that further acquisitions can continue.
- The proposed method is highly efficient as it processes an active pool of size just $m * p$ points per iteration, where m is the number of clusters and p is a user defined cluster batch size.
- The proposed method is robust to added noise in the images.

The next section describes various active learning algorithms in the literature. The following section, describes our proposed method *Diverse Bayesian Active Learning with Simulated Annealing*. The performance of the proposed method is compared with some of the state of the art methods in the *Experiments* section. This section also describes the implementation details of all these methods. Finally, the last section concludes the chapter.

2.2 RELATED WORK

Based on existing literature (Settles, 2010), active learning may query unlabeled data points in one of the three ways -

Membership-query synthesis. This kind of active learning method is generative in nature. In every iteration, the algorithm generates a batch of samples within the input domain, which can be sent to the oracle for labeling. The algorithm can be designed to generate informative samples. However, the generated samples might be semantically difficult to label especially for a human oracle.

Stream based. In this kind of querying setup, the active learner is provided with a continuous stream of unlabeled samples, one at a time. The active learner has to decide whether to send the current sample for labeling. This kind of setup is particularly useful in stochastic processes or agent based learning but highly ineffective for deep learning.

Pool based. In this setup, the active learner has access to the entire unlabeled pool of samples. In each iteration, the active learner has to acquire a single or a batch of informative samples from the unlabeled pool. The active learner may process the whole pool or a part of the pool in each iteration depending on its size and computational resources.

In this thesis, we employ a pool based setup which is very common in the real world. This is due to the increasing availability and affordability of datasets and computational resources. Further, we focus on active learning methods relevant for image classification tasks. They are not necessarily adaptable for semantic segmentation tasks. We discuss about active learning for semantic segmentation in the next chapter. Based on the strategy used for acquisition, we divide active learning methods as follows (see figure 2.1)-

Uncertainty based. Most common active learning methods are uncertainty based that use uncertainty as a measure of informativeness. In shallow machine learning methods such as SVM, the distance from decision boundary is used as the uncertainty measure (Tong & Koller, 2002). This is also known as margin based active learning. However, computing decision boundaries for ensemble or deep networks is intractable. As a result, metrics such as entropy and margin score are used to evaluate uncertainty in such models. These metrics are calculated using the network predictions. Ensemble based active learning, also known as query-by-committee active learning method where modeling uncertainty is trivial. As fas as deep learning is concerned, ensemble learning is computationally expensive. Therefore, variational inference (see A) and bayesian statistics are used to model uncertainty in deep networks. The earliest work of Kapoor et al., (2007) used a gaussian process prior for probabilistic object classification. Houlsby et al., (2011) proposed an acquisition function BALD (Bayesian Active Learning by Disagreements) that uses gaussian process to model the mutual information between posterior distribution of model and its prediction. CEAL (Cost Effective Active Learning) promised to improve the network's generalization by acquiring high confidence samples through pseudo labeling in addition to uncertainty sampling (K. Wang et al., 2017). However, pseudo labeling introduces noise into the training set when hyperparameters are poorly tuned. Gal et al., (2017) proposed a bayesian active learning framework DBAL (Deep Bayesian Active Learning) based on MC dropouts that can be combined with a variety of acquisition functions. Woo et al., (2021) proposed an acquisition function BABA (Beta Approximation for Bayesian Active Learning) which is a normalized version of BALD. In BABA, class probabilities are approximated using a beta distribution and is used for calculating the normalising term for BALD.

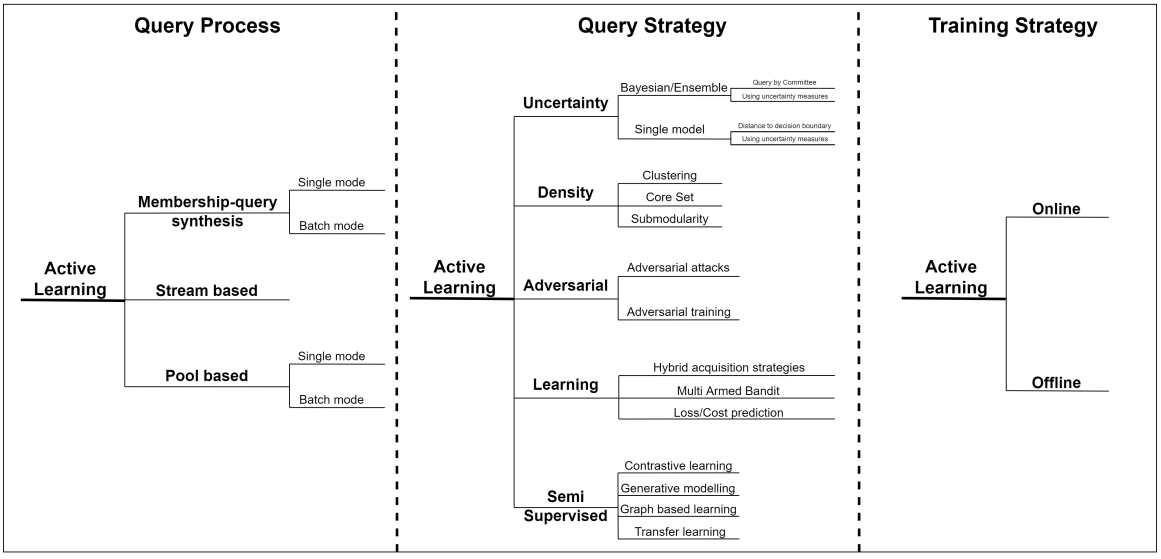


Figure 2.1: Categorization of active learning methods.

Despite the initial success of these uncertainty based methods, it was discovered that they fail to capture diversity among samples in a batch mode setting, especially in complex datasets. Huang et al., (2014) combined uncertain and diverse sampling into a min-max optimization problem. But, they only focused on binary classification problems and non deep learning methods. Recently, Pop et al., (2018) extended DBAL to address mode collapse in complex multi-class problems through ensemble learning. However, ensemble learning is not suitable for deep networks as mentioned before. BatchBALD (Kirsch et al., 2019) extended BALD for batch mode acquisitions. Here, diversity is measured by the joint entropy between batch of samples and model predictions. Similar to batchBALD, BABA is easily extended to batch mode. However, these methods can only work on small active pools and become infeasible for large acquisition sizes of the order of 1k[1].

Density based. These methods are specifically designed for batch mode acquisitions. These methods trade off informativeness and representativeness by learning the data distribution along with the structure of data (Kim et al., 2021). Initial approaches to density based methods viewed batch mode active learning as a subset selection problem. For instance, Wei et al., (2015) combine uncertainty sampling with data subset selection using submodularity to data likelihood functions. Core Set approach proposed by Sener et al., (2018) reduces the unlabeled pool into several subsets of diverse points. They use mixed integer programming to find these optimal subsets. Here, informativeness is traded with representativeness by ignoring training and generalization loss in the overall active learning loss. As the approach becomes infeasible for large datasets, the authors suggest the use of greedy 2-approximation of the k-center problem.

Recent approaches to batch mode active learning use clustering to diversify uncertain samples. In Zhdanov, (2019), diversity is captured by acquiring samples close to KMeans cluster centroids and uncertainty is captured by adding weights to KMeans clustering. But it requires to run KMeans in every iteration which is a hugh bottleneck. In Ash et al., (2019) (BADGE), authors argue that most informative samples should have large corresponding gradients. They combine diversity and uncertainty by using KMeans++ seeding algorithm on the gradient of penultimate layer of the network. However, BADGE becomes infeasible for large active pools and it doesn't scale well with the number of classes. More recently, Citovsky et al., (2021) proposed cluster based diversification

[1]We verify this fact through experimentation on cifar10 with acquisition batch size of 1k.

of uncertain samples (Cluster Margin). They first perform clustering on the embedding of the penultimate layer of the network. They generate the active pool by selecting samples with the lowest margin scores. They then reduce the active pool to required batch size through round robin sampling from clusters starting from the smallest cluster. The potential downside of this method is that it is infeasible to compute the margin scores for entire pool in each iteration especially in large datasets. Further, the optimal size of the active pool remains unclear. This is a direct result of sample-then-cluster strategy.

Adversarial based. Yet another class of active learning methods use adversarial training either for training a separate network or generate adversarial samples. Duocoffe et al., (2018) use deep fool algorithm to generate adversarial examples and select those samples which have the smallest perturbation from their corresponding adversarial example. They also add the corresponding adversarial examples into the training set to regularize the network training. In (Sinha et al., 2019), variational autoencoder is used to learn a latent space for the labeled and unlabeled pool. Then a discriminator network is used to distinguish between labeled and unlabeled samples and those unlabeled samples are acquired which have the highest uncertainty with respect to the discriminator. However, both these methods do not explicitly model diversity in batch mode setting. Liu et al., (2020) have presented an adversarial active learning method for outlier detection. They use GAN to generate outliers and tackle the problem of high dimensionality. It is worth noting that most of these methods are only successful in certain experimental settings. These methods can easily be affected by high intra class variance in the dataset. Also, the initial network may be highly biased for adversarial training due to lack of enough training data.

Learning based. Learning based active learning methods try to learn or model the most optimal acquisition function. Hsu et al., (2015) presented a multi armed bandit problem which chooses the most optimal active learning strategy by using the network's performance as a feedback. Yoo et al., (2019) argue that informative samples have large corresponding losses. They construct a loss prediction module which can predict losses for unlabeled data points. The major drawback of this method is that diversity is not considered among the acquired data points. Kushnir et al., (2020) has presented a exploration versus exploitation view of active learning. The exploration stage of active learner tries to model the data distribution while its exploitation stage models the structure of the data.

Semi Supervised. Lastly, active learning methods can also use semi supervised techniques such as contrastive learning (Margatina et al., 2021) or generative modelling (Tran et al., 2019) to acquire samples. These methods are sophisticated and less relevant to our work and hence only briefly mentioned here.

2.3 ALGORITHM

2.3.1 Notation and Setting

Active learning starts with a feature space $\chi \in \mathbb{R}^{nxmxc}$. The label space $Y \in \{1, 2, ...K\}$ is generated by an oracle S given χ. We consider a pool based batch mode active learning approach which consists of an unlabeled set $U \sim \chi$, a labeled set $L = \{(x_i, y_i) \mid x_i \in \chi, y_i \in Y\}$ and a network C with initial parameters θ. We denote the acquisition function as Q. The acquisition cost generated by Q is denoted by E and the batch acquisition size per iteration is denoted by k. Furthermore, the

size of the active pool generated from U is denoted as p. Most often, C is pretrained on randomly selected samples from U to ensure unbiased evaluation of Q.

2.3.2 Batch Mode Active Learning

The goal of batch mode active learning (BMAL) is to select a batch of samples in a single iteration. This helps in faster and better convergence of deep networks. It also reduces running time as network is trained less frequently as compared to traditional active learning. The BMAL problem is defined as -

$$\{x_1^*, x_2^*..., x_k^*\} = \underset{A \subset \{x_1, x_2, ...x_p\}, |A|=k}{\text{argmax}} Q(\{x_1, x_2, ...x_p\}, C) \tag{2.1}$$

Algorithm 1 Batch Mode Active Learning

Require: L, U, Q, S, C, k, p
Perform Pre-Train C on L
while not **done**:
  Sample $\{x_1, x_2, ...x_p\} \in U$
  $L^* \leftarrow \{x_1^*, x_2^*..., x_k^*\} \leftarrow \underset{A \subset \{x_1, x_2, ...x_p\}, |A|=k}{\text{argmax}} Q(\{x_1, x_2, ...x_p\}, C)$
  Label L^* using S
  $L \leftarrow L \cup L^*$
  Train C on L
end while

BMAL presented in Algorithm 1 acquires k samples in each iteration processing p samples randomly sampled from U. The most optimal Q for BMAL is one which evaluates all possible ${}^pC_k$ subsets. This is in most cases computationally infeasible.

2.3.3 Bayesian Active Learning with Simulated Annealing

Active learning with simulated annealing (ALSA) presented in Algorithm 2, starts by clustering the pool U into m clusters. We propose the use of KMeans clustering as it is fast and scalable for large datasets (James & others, 1967). We maintain the acquisition cost for each cluster in the array E_m. To make the algorithm more efficient, we set the initial cost of each cluster to e rather than 0. We use a diversity measure along with an uncertainty measure for the acquisition cost. The diversity measure of a sample x_i is calculated by its L-2 norm from its cluster centroid c_i given by:

$$D(x_i) = ||x_i - c_i||_2 \tag{2.2}$$

We use the margin score as the uncertainty measure which is calculated for a sample x_i after averaging the class probabilities of its N MC samples given by:

$$Z(x_i, \theta) = P_\theta(y_1 \mid x_i) - P_\theta(y_2 \mid x_i) \tag{2.3}$$

Algorithm 2 Active Learning with Simulated Annealing

Require: e, α, β, n_{epochs}, m, k
Set $T \leftarrow 1$, $E_m \leftarrow e$, $L \leftarrow \phi$, $B \leftarrow \phi$
Perform KMeans Clustering with m clusters
for $n = 1$ to n_{epochs} **do**:
  Sample $\{{x_1}^1, {x_2}^1, ...{x_p}^m\}$ (p from each cluster)
  for each $\{{x_1}^1, {x_2}^1, ...{x_p}^m\}$:
    #Compute Uncertainty using MC Dropout
    #Compute L-2 norm from cluster center
    ${E_j}^*(x_i^j) \leftarrow T * D({x_i}^j) + (1 - T) * Z({x_i}^j, C(\theta))$
    if ${E_j}^* > E_j$ **then**:
      $E_j \leftarrow {E_j}^*(x_i^j)$
      Label ${x_i}^j$ using S
      $B \leftarrow B \cup \{{x_i}^j, {y_i}^j\}$
    else if $randu(0, 1) < e^{-\alpha * T(n) * (E_j - {E_j}^*(x_i^j))}$ **then**:
      $E_j \leftarrow {E_j}^*(x_i^j)$
      Label ${x_i}^j$ using S
      $B \leftarrow B \cup \{{x_i}^j, {y_i}^j\}$
  end for
  $T \leftarrow e^{-\beta \frac{n}{n_{epochs}}}$
  if $|B| >= k$ **then**:
    $L \leftarrow L \cup B$
    Train $C(\theta)$ on L
    $B \leftarrow \phi$
end for

where y_1 and y_2 are the two largest class probabilities. Diversity is enforced by selecting samples with the *least* L-2 norm from their cluster centroids and uncertainty is enforced by selecting samples with the *least* margin scores. We use margin score as it ranges between 0 and 1, and normalise the L-2 norm to 0-1 range using min-max scaling.

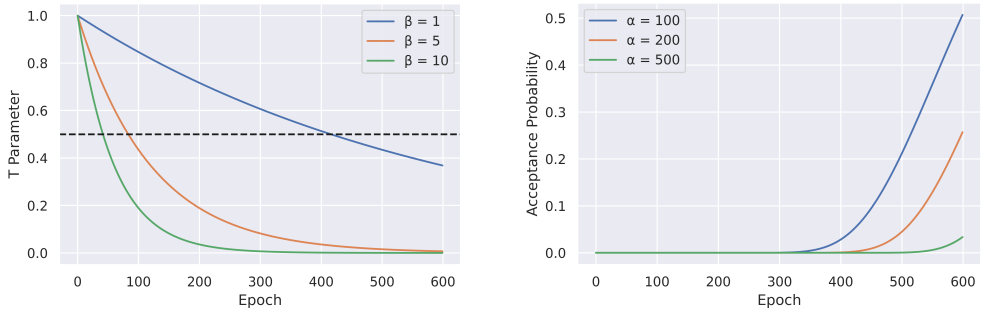
When it comes to the acquisition strategy, we propose to select diverse samples in the beginning so that sufficient distribution of the data is learned by the network before enforcing uncertainty. This solves the initial seed set problem and allows for less biased uncertainty estimations (Farquhar et al., 2021; Xu & Mannor, 2010). To achieve this, we propose the use of simulated annealing. We create a convex combination of D and Z using temperature parameter T which varies from 1 to 0 given by:

$$E_j^*(x_i^j) = T * D(x_i^j) + (1 - T) * Z(x_i^j, \theta) \tag{2.4}$$

where x_i^j is the i^{th} sample from j^{th} cluster. The annealing schedule of T follows a negative exponential function given by:

$$T(n; \beta, n_{epochs}) = e^{-\beta \frac{n}{n_{epochs}}} \tag{2.5}$$

where n is the current epoch number, n_{epochs} is the total number of epochs and β is a hyperpa-



(a) (b)

Figure 2.2: (a) Variation of T parameter over epochs with 3 different values of β; (b) Variation of acceptance probability over epochs with 3 different values of α [when $\beta = 5$ and $E_j - E_j^* = 1$].

rameter. Notice that the final cost ranges between 0 and 1 and gradually changes from diverse to uncertainty sampling. We follow a cluster-then-sample strategy by generating the active pool by selecting p samples at random from each cluster. A sample is acquired only if its current cluster cost E_j is less than the sample's cost. If acquired, the cluster cost is updated with the sample's cost. This kind of acquisition allows for a not so greedy selection of samples. If the sample is not acquired, it may be acquired through an acceptance probability. We propose an *inverse acceptance probability* criteria wherein the probability increases with decrease in T given by:

$$P_n(E_j^*(x_i^j) \mid E_j, T(n); \alpha) = e^{-\alpha * T(n) * (E_j - {E_j}^*(x_i^j))} \tag{2.6}$$

where α is a hyperparameter. This ensures that the algorithm is more careful in selecting the samples in the beginning. Also, it allows the algorithm to escape maxima of acquisition costs. Note that the proportion of diverse to uncertainty sampling can controlled by varying the β hyperparameter. The effect of α on acceptance probability and β on T is presented in the figure 2.2. Through rigorous experiments, we found optimal values of α and β to be 200 and 5 respectively.

2.4 EXPERIMENTS

We assess the performance of our proposed method on three benchmark datasets commonly used in the active learning literature i.e. *Mnist* (LeCun et al., 2010), *Fashion Mnist* (Xiao et al., 2017) and *Cifar10* (Krizhevsky & Hinton, 2009) (see D). We use the *LeNet5* (LeCun et al., 1998) CNN architecture for all the evaluations (*epochs=50, Adam, lr=0.001, batch=128*). Additionally, two dropout layers (p=0.25, p=0.55) are added after the first convolutional and linear layer respectively for drawing approximate bayesian inferences. The acquisition batch sizes and the sizes of the initial seed set are - 32, 32, 1000 respectively for the datasets. Also note that ALSA does not require the network to be pretrained on an initial seed set as its initial diverse sampling stage is independent of network predictions.

2.4.1 Comparison with baselines

The results achieved by our proposed method on the test set are compared with the following baseline methods:

1. **Cluster Margin** (Citovsky et al., 2021): We perform hierarchical agglomerative clustering (HAC) on the embedding of penultimate layer of the pre-trained network. We select top 5000 samples from the entire pool which have the lowest margin scores. We then select k samples from the active pool by selecting one sample at random from each cluster in a round robin fashion starting from the smallest cluster.

2. **BADGE** (Ash et al., 2019): We select k samples from randomly generated active pool of 5000 samples using KMeans++ seeding algorithm on the gradient embedding of the samples. Gradient embedding is calculated by the gradient of the loss on the penultimate layer.

3. **BALD** (Houlsby et al., 2011): We select the top k samples from the entire pool which have the highest mutual information among the classifiers. We use 10 MC samples to model uncertainty.

4. **Core Set** (Sener & Savarese, 2018): We select k samples from the entire pool which form a core set by solving the k-center problem on the labeled and unlabeled pool. To keep the evaluations computationally comparable, we use the greedy 2-approximation solution to the k-center problem as suggested by the authors.

5. **DBAL** (Gal et al., 2017): We select the top k samples from the entire pool which have the highest entropy. We use 10 MC samples to model uncertainty.

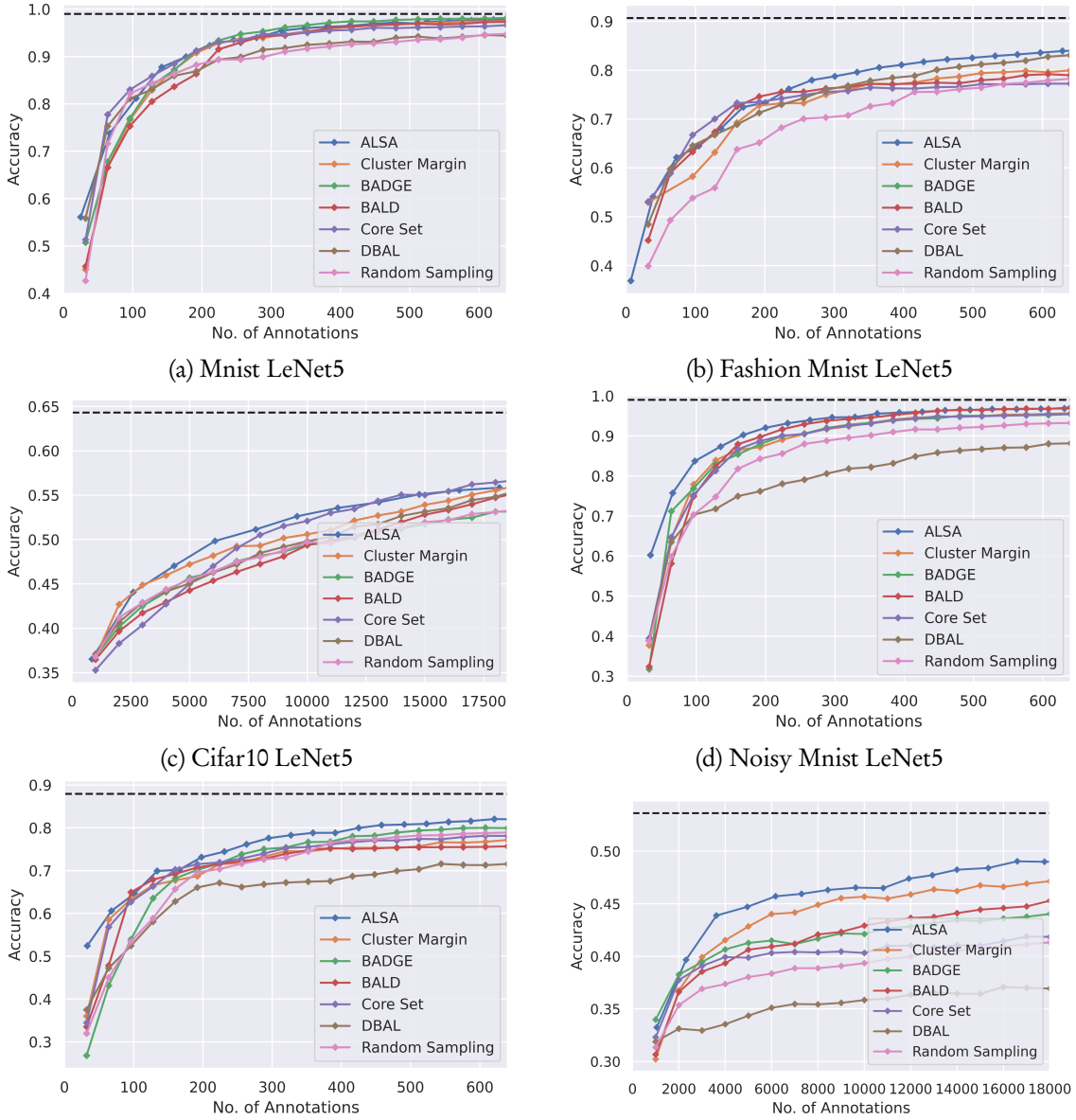
6. **Random Sampling**: We randomly select k samples from the entire pool.

We report the results by taking an average over 3 runs. To keep the evaluations identical, dropout layers are still included for non-BNN methods. For ALSA, we use 10 MC samples and 10 clusters for every experiment.

Firstly, we observe that all the active learning methods perform better than or at least identical to random sampling (Figure 2.3). We note that DBAL performs identical to random sampling on *Mnist* and *Cifar10* and BALD performs identical to random sampling on *Cifar10*. In general, BALD and DBAL are the lowest performers. This is due to the lack of a diversity measure when acquiring samples in a batch mode (see figure 2.4). Independent of the dataset, our method consistently performs better than most of the active learning methods.

In the initial set of acquisitions, ALSA performs similar to other active learning methods. However, it converges faster than rest of the methods in subsequent acquisitions by a large margin (Table 2.1). This confirms our initial hypothesis that careful selection of the initial seed set leads to a better performance with fewer queries. Interestingly, BADGE outperforms ALSA on *Mnist* by a small margin. This is owing to the fact that *Mnist* is a less challenging dataset as compared to the other two and the random nature of ALSA. When it comes to Cluster Margin, its performance is highly dependent on the generation of optimal clusters and the active pool. We noted that tuning the distance threshold ϵ and the size of active pool is quite challenging and the tuning has to be performed separately for each experiment. Unlike Cluster Margin, BADGE performs KMeans++ seeding on the entire pool instead of top p uncertain samples. This means that the algorithm also selects high confidence samples which may prove to be suboptimal during final set of acquisitions.

Table 2.2 compares the runtime of active learning methods for first 20 queries on *Mnist*. As expected, non-density based methods DBAL and BALD have the lowest runtimes. On the other hand, ALSA has the lowest running time as compared to other diversity based methods: Cluster Sampling, BADGE and Core Set. It should be noted that the runtime of BADGE increases with



(a) Mnist LeNet5

(b) Fashion Mnist LeNet5

(c) Cifar10 LeNet5

(d) Noisy Mnist LeNet5

(e) Noisy Fashion Mnist LeNet5

(f) Noisy Cifar10 LeNet5

Figure 2.3: Classification accuracy (on test set) of various active learning algorithms as a function of number of annotated images on the three datasets using LeNet5.

increase in the size of active pool and the dimension of embedding layer. The runtime of Core Set increases with increase in the size of the labeled set. On the other hand, the acquisition operation of ALSA is independent of these parameters. The largest bottleneck for Cluster Margin is the HAC step.

2.4.2 Effect of Noise

Apart from data availability, data quality is also a prime requirement for the success of deep learning applications. There are very few studies that analyse the effect of noisy features on active learning

Table 2.1 Classification accuracy (on test set) of various active learning algorithms with number of annotated images on the three datasets using LeNet5. *For ALSA, the number of annotations are the maximum value less than or equal to the given size of annotated set.

(a) Mnist LeNet5

# Annotations*	160	320	480	640
ALSA	**89.92**	95.97	96.98	97.89
Cluster Margin	87.23	94.70	96.70	97.47
BADGE	87.15	**96.15**	**97.72**	**98.15**
BALD	83.63	94.45	96.78	97.29
Core Set	88.46	94.68	95.99	96.61
DBAL	85.93	91.79	93.94	94.40
Random	86.34	91.00	93.08	94.77

(b) Fashion Mnist LeNet5

# Annotations*	160	320	480	640
ALSA	68.10	**78.76**	**82.21**	**83.93**
Cluster Margin	72.76	77.20	79.41	80.11
BADGE	68.87	76.83	80.70	83.08
BALD	72.57	76.53	77.32	78.97
Core Set	**73.29**	75.74	76.59	77.24
DBAL	70.95	77.92	79.92	81.07
Random	63.78	70.73	76.11	78.26

(c) Cifar10 LeNet5

# Annotations*	1000	5000	10000	15000
ALSA	36.53	47.03	**52.62**	**55.09**
Cluster Margin	36.79	**47.20**	50.59	53.91
BADGE	35.87	46.32	51.03	53.83
BALD	36.47	44.27	49.36	52.80
Core Set	35.28	45.00	52.09	55.02
DBAL	**37.10**	45.06	49.81	53.14
Random	36.79	45.47	49.72	51.95

(d) Noisy Mnist LeNet5

# Annotations*	160	320	480	640
ALSA	**90.25**	**94.69**	96.47	96.85
Cluster Margin	86.49	92.39	94.87	95.67
BADGE	85.36	92.78	95.03	95.42
BALD	87.91	94.23	**96.55**	**97.10**
Core Set	86.65	92.55	94.77	95.45
DBAL	74.95	81.81	86.33	88.14
Random	81.78	89.54	92.02	93.24

(e) Noisy Fashion Mnist LeNet5

# Annotations*	160	320	480	640
ALSA	70.21	**78.27**	**80.77**	**81.96**
Cluster Margin	67.72	74.54	75.30	77.16
BADGE	68.19	75.42	78.88	79.92
BALD	69.22	73.93	75.42	75.67
Core Set	**70.23**	75.28	77.05	78.11
DBAL	62.77	67.20	69.91	71.56
Random	65.67	73.03	77.81	78.90

(f) Noisy Cifar10 LeNet5

# Annotations*	1000	5000	10000	15000
ALSA	33.24	**45.71**	**47.39**	**48.99**
Cluster Margin	30.22	42.84	45.67	46.75
BADGE	**33.98**	41.27	42.12	43.35
BALD	30.66	40.63	42.92	44.44
Core Set	32.30	39.87	40.32	41.02
DBAL	31.87	34.35	35.84	36.44
Random	31.34	38.03	39.35	40.79

Table 2.2 Average runtime of active learning methods on Mnist for first 20 acquisitions. The runtimes for ALSA and Cluster Margin are inclusive of their initial clustering step.

Method	Runtime (Seconds)
ALSA	173.12
Cluster Margin	590.15
BADGE	381.98
BALD	95.92
Core Set	465.25
DBAL	**93.56**

(Abraham & Dreyfus-Schmidt, 2021). Noisy features are very common in the real world which may be a direct result of errors in collecting training data or errors introduced during transmission of features to the oracle. In this experiment, we study the performance of active learning methods where 50% of the unlabeled pool U is corrupted by Gaussian noise. We corrupt only half of the

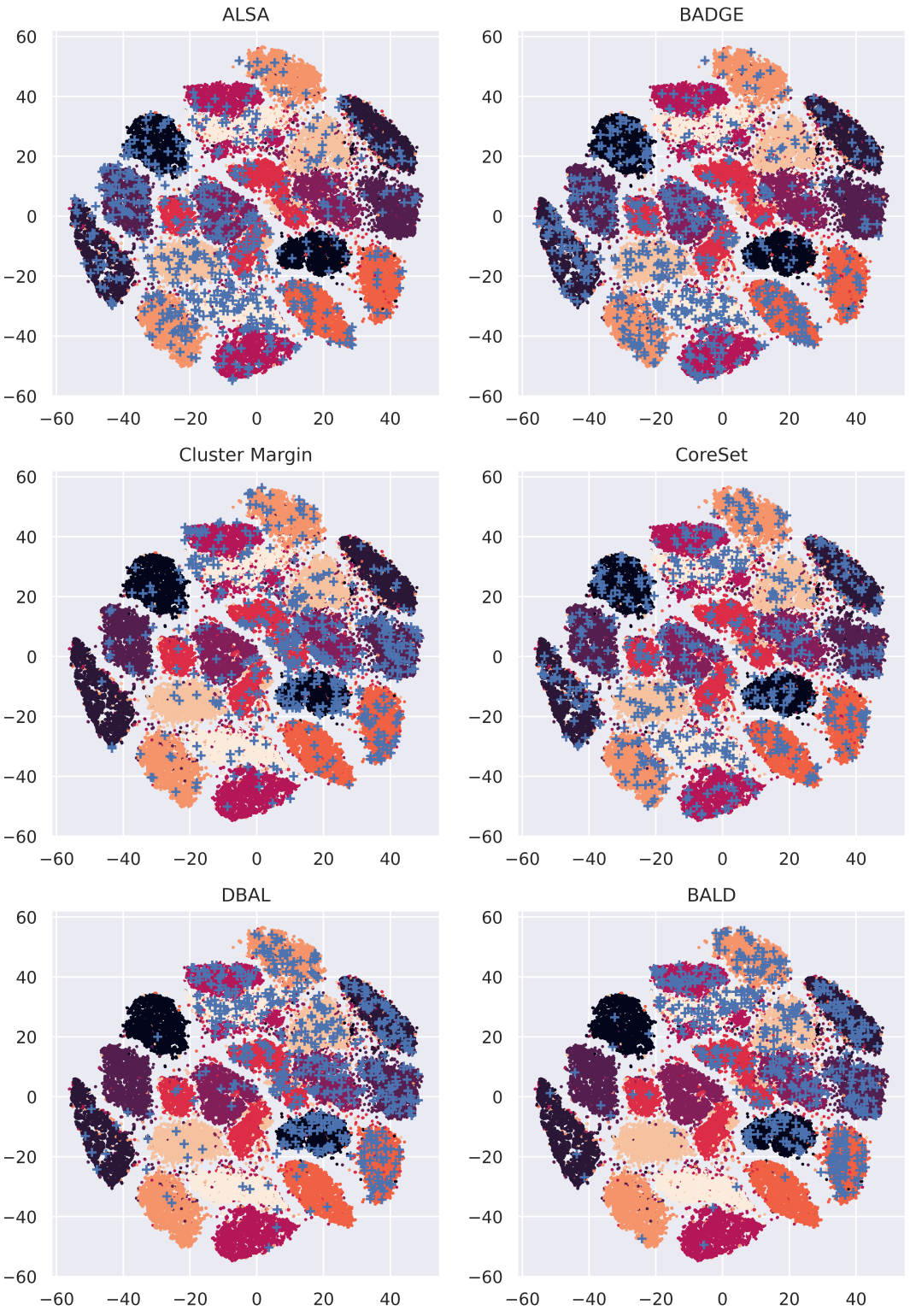


Figure 2.4: tSNE embeddings of the Mnist dataset and the images acquired by the active learning methods. Density based active learning methods acquire images evenly across the feature space.

pool U to study the ratio of good quality samples to noisy samples (S/N) acquired by each active learning method. We employ the *Berkson error model* as described in (Ramdas et al., 2014) but with Gaussian noise given by:

$$U^* = U + \mathcal{N}(0, 1) \tag{2.7}$$

Noise is added to the images after normalising them between 0 and 1. Furthermore, the noisy pixel values are clipped between 0 and 1 to avoid out of domain input to the network. Furthermore, we assume an ideal oracle that correctly labels all the noisy features without any error.

Table 2.3 The ratio of good quality samples to noisy samples (S/N) acquired by the active learning methods. Some active learning methods achieve sufficient accuracy even with poor S/N ratio as noisy samples regularizes the network training.

METHOD	S/N RATIO		
	MNIST	FASHION MNIST	CIFAR10
ALSA	***3.75***	***2.12***	2.23
CLUSTER MARGIN	1.47	2.07	2.21
BADGE	0.72	0.96	0.92
BALD	2.67	0.72	***2.29***
CORE SET	0.84	1.46	0.62
DBAL	0.21	0.29	0.39
RANDOM	1.06	1.10	0.97

Table 2.3 compares the ratio of good samples to noisy samples acquired by the active learning methods. One can argue that noisy images represent the highest uncertainty. As a result, DBAL is the worst performing algorithm as it selects the images with the highest entropy over the class probabilities. BALD is able to select more of the good quality images as it selects images based on the mutual information between the predictions and network parameters. In general, we observed low BALD scores for noisy images. Among the density based methods, only ALSA and Cluster Margin are able to clearly outperform random sampling on all the three datasets (Table 2.1). As noisy images represent high variance, Core Set fails to perform well in this experiment. BADGE also performs poorly as noisy images output large gradient embeddings. Independent of the dataset, ALSA selects at least twice the number of good quality images as the number of noisy images. This is because noisy images lie close to the cluster boundaries and the diversity measure of ALSA makes it select images close to the cluster centers. Another advantage of ALSA when applied to noisy datasets is that its diverse sampling stage can be enhanced by decreasing the value of β. In general, ALSA can be adapted to any dataset by effectively varying the hyperparameters α and β.

2.4.3 Ablation Study

We investigate the performance of ALSA under certain ablations. Particularly, we intend to investigate the contribution of simulated annealing and the diversity measure towards an optimal acquisition of samples. We compare the performance of ALSA against the following ablated models:

1. **UALSA**: We remove the diversity measure so that the acquisition cost only consists of the margin score. The simulated annealing procedure selects those samples which improve the

margin score of samples acquired in the previous iteration. Also, the model is pretrained on 32 randomly selected samples for subsequent uncertainty evaluations.

2. **Cluster Sampling**: We remove simulated annealing and the diversity measure and we select top 4 uncertain samples from each cluster. The active pool is generated by randomly sampling 100 images from each cluster. With 10 clusters, the acquisition size is 40. Also, the model is pretrained on 32 randomly selected samples.
3. **Random Pool ALSA**: We generate the active pool by randomly sampling 32 images from the entire pool instead of sampling from the clusters. Rest of the procedure is followed as in ALSA.
4. **ALSA-NonBNN**: We remove the MC dropout component from the overall ALSA framework to study the effect of approximate bayesian inference for modeling uncertainty.
5. **ClusterMargin-BNN**: We introduce MC dropout to model uncertainty and compute margin sampling scores in the ClusterMargin framework.

Table 2.4 Classification accuracy (on test set) of various ablated model with number of annotated images on Mnist using LeNet5.

# ANNOTATIONS*	160	320	480	640
ALSA	89.92	***95.97***	***96.98***	***97.89***
UALSA	88.21	94.19	95.75	96.45
CLUSTER SAMPLING	82.83	92.64	94.49	95.52
RANDOM POOL ALSA	81.08	93.33	94.62	95.06
ALSA-NONBNN	***92.44***	95.36	96.09	96.26
CLUSTER MARGIN	87.23	94.70	96.70	97.47
CLUSTERMARGIN-BNN	89.86	94.37	95.70	96.64

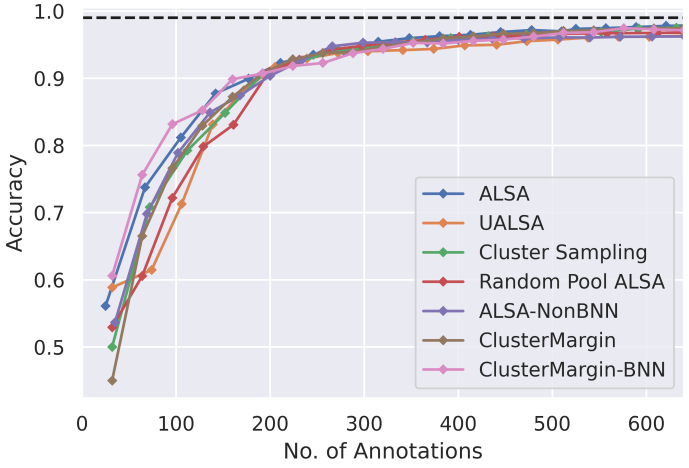


Figure 2.5: Comparison of Classification accuracy (on test set) of ablated models with ALSA as a function of number of annotated images on the three datasets using LeNet5.

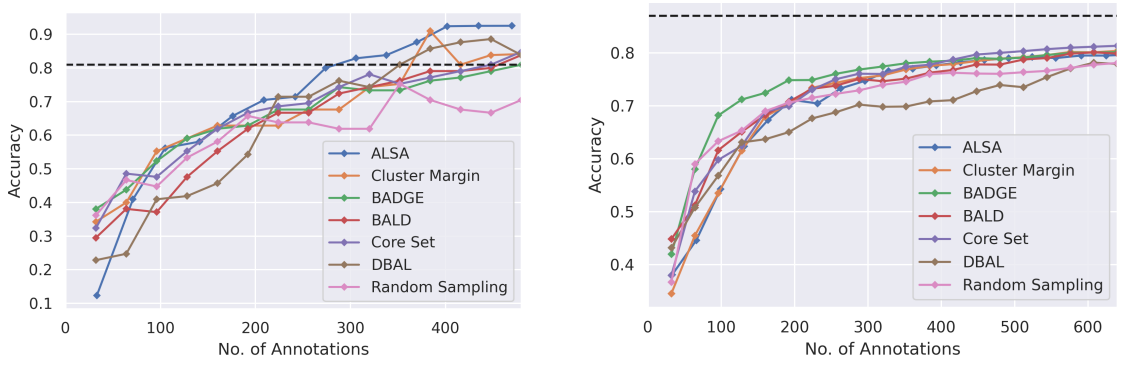
We note that UALSA achieves performance that is very close to ALSA (see Figure 2.5, Table 2.4. Its lower performance is due to the lack of the diversity measure which makes it acquire samples

that are close to the cluster boundaries. Both, Cluster Sampling and Random Pool ALSA perform poorly as compared to both ALSA and UALSA. The difference between UALSA and Cluster Sampling is that the later selects highly uncertain samples from a randomly generated active pool within a cluster. As a result, its performance is affected in iterations where poor quality of active pool is generated. The simulated annealing procedure in UALSA solves this problem by selecting those uncertain samples which improve the acquisition cost of acquired samples in the previous iteration. Random Pool ALSA is the lowest performer among the models. Even with the diversity measure and simulated annealing, it is not able to select distinct samples leading to mode collapse. This clearly depicts the importance of careful selection of the active pool. Similar observations are made when training the models on *Fashion Mnist* and *Cifar10*.

2.4.4 Application on Remotely Sensed Images

We evaluate our method on benchmark image classification datasets in remote sensing i.e. *UC-Merced* (Yang & Newsam, 2010), and *EuroSat* (Helber et al., 2019). Due to the complexity of the datasets, we use the VGG16 CNN architecture for classification. We use a pretrained version of VGG16 on imagenet. We only train the head of the network which consists of two fully connected layers. We apply dropout to both fully connected layers for approximate bayesian inference. Due to the smaller size of the datasets, we set the active pool size for BADGE and Cluster Margin to 1000.

Firsty, we notice that the accuracies achieved by most of the active learning methods over the course of iterations are not monotonically increasing. In fact, the training and validation loss curves are not as stable as was with previously experimented datasets. The general observation was that adding more labeled images into the existing training data did not necessarily guarantee increased accuracy. This is owing to the fact that remote sensing datasets are challenging to classify (in general) due to the presence of noise and high variance. Interestingly, ALSA is able to achieve an accuracy of 92.38% on *UC-Merced* with just 402 annotated images, where the baseline model is able to achieve 80.95%. Other active learning methods are also able to barely outperform the baseline model at around the same number of images in the training dataset as in ALSA. The prime reason of such an outcome is the poor capacity of the VGG16 network. This network is not complex



(a) UC Merced VGG16 (b) Eurosat VGG16

Figure 2.6: Classification accuracy (on test set) of ALSA as a function of number of annotated images on remote sensing datasets.

enough to represent the variance in the entire *UC-Merced* dataset. However, with a subset of the dataset, it is able to fit and generalize well. Active learning methods perform similarly on *EuroSat* with the exception of DBAL which performs worse than random sampling.

2.5 CONCLUSIONS

In this chapter, we presented a novel framework for combining simulated annealing with active learning. This approach allows for less greedy selection of samples and is highly efficient. We also illustrate the competitive results of our approach using Bayesian CNNs against state of the art active learning methods. In general, density based methods outperformed uncertainty based methods. Our framework is agnostic in nature in the sense that it can be combined with any acquisition function and any ensemble or Bayesian machine learning algorithm. The potential downside of this framework is that its clustering step could prove to be a bottleneck in case of large datasets. But this can be easily addressed using approximation algorithms such as KMeans++ seeding algorithm. In the future, this framework can be evaluated using clustering techniques other than KMeans and distance indexes other than L-2 norm.

Chapter 3

Active Learning for Semantic Segmentation

> Nothing can add more power to your life than concentrating all of your energies on a limited set of targets.
>
> Nido Qubein

3.1 INTRODUCTION

One important task in computer vision and remote sensing is per-pixel labeling of images. This is paramount in computer vision applications such as object localization and 3D reconstruction. Semantic segmentation (per-pixel classification) is arguably the most important task in remote sensing. It has diverse applications ranging from crop monitoring to analysing the impact of natural hazards. Being task-specific, supervised segmentation approaches (almost always) outperform unsupervised approaches. Fully Convolutional Neural Networks (FCN) have become the standard in most supervised segmentation tasks. FCN's are more efficient than vanilla CNN's at segmentation as a CNN requires us to iterate at pixel level for classification. On the other hand, FCN can provide labels to a patch of pixels at once. However, their performance is mainly dependent on the availability of labeled images. Inadequacy of labeled images becomes a bottleneck especially when using overparameterized FCN's.

Labeled training images are usually produced by expert human annotators. In computer vision applications, images are annotated using annotation tools such as LabelMe (Torralba et al., 2010). In remote sensing, the annotation process can be carried out through: 1) ground surveys; 2) imagery interpretation. The label assignment is usually done in GIS software. Independent of the application, it is time consuming and expensive to obtain large amounts of labeled images for training a deep learning network. Active learning can be used to reduce the efforts of human annotators. The aim of active learning is to reduce the number of labeled images required while maintaining the performance of deep learning models at an acceptable level.

In the previous chapter, we discussed the application of deep active learning for image classification. The definition of active learning for image classification cannot be adopted directly for semantic segmentation. Unlike image classification, semantic segmentation requires network to provide label to each pixel in a given image. The fundamental questions here are: 1) whether the active learning algorithm be designed to acquire individual pixels or a regular patch of pixels?; 2) should the active learning algorithms be evaluated at pixel-level or patch-level?; 3) how to compare

active learning algorithms that are designed for pixel-level with those designed for patch-level acquisitions?

It is feasible to acquire individual labeled pixels when using shallow machine learning algorithms for segmentation. However, this approach is highly inefficient for deep active learning methods that use FCN like architectures for segmentation. Segmentation of large scale images (with FCN) is usually performed by first dividing them into small rectangular patches of pixels of equal sizes and then classifying each of them. The nature of this approach constraints active learning algorithms to acquire patches rather than individual pixels. Further, these methods cannot be directly compared with active learning methods that acquire pixels. In this work, we primarily focus on deep active learning methods for which acquiring patches is the most optimal.

One of the key challenges is to define the informativeness of images in case of semantic segmentation. For segmentation, images need to be provided labels at a pixel-level. As a result, uncertainty evaluations also need to be performed at the pixel-level. Most of the current active learning methods use average predictive uncertainty of pixels within a patch as a metric for acquisition. However, this may be highly biased considering the variability and distribution of classes within a patch. Second, clustering of patches in case of density based methods is not so trivial. Methods such as CoreSet, which require the computation of distance between images also cannot be directly applied for segmentation tasks. For such methods, the patches first need to be embedded into a latent space for computing distances or performing clustering.

In remote sensing, spatial autocorrelation introduces new challenges to active learning. The accuracy achieved on validation/test site is highly dependent on its spatial autocorrelation with the training site. If the validation/test site is not well separated from the training site, then acquiring training patches close to the validation/test site would produce maximum accuracy. Active learning shall not be useful in such scenarios. As a result, careful selection of the training, validation and testing set need to done before evaluating active learning algorithms. Due to this, active learning for semantic segmentation automatically translates to the problem of spatial transferability in remote sensing.

In this work, we propose an active learning method that is effective for spatial transferability. The predictive uncertainty evaluation is performed at a class level rather than at the patch level. The predictive uncertainty of each class is weighted by its proportional distribution. This metric is calculated for both the training and the testing patches. The training patches are acquired based on their similarity of the uncertain class distribution with the testing patches. The contribution of this work is threefold:

- The proposed method solves the problem of biased evaluation of acquisition function at patch level.
- The proposed method is effective for spatial transferability of deep learning models.
- The proposed method acquires those training patches that have similar class distribution as in the testing site and rejects irrelevant classes for acquisition.

The next section describes various perspectives of active learning for semantic segmentation in the literature. The following sections, describe the study area and dataset considered and our proposed method *Active Learning by Uncertain Class Diversity Conditioned on Target Site*. The

performance of the proposed method is compared with some of the state of the art methods in the *Experiments* section. This section also describes the implementation details of all these methods. Finally, the last section concludes the chapter.

3.2 RELATED WORK

3.2.1 Semantic Segmentation

A lot of studies have proposed supervised segmentation approaches in the past (L.-C. Chen et al., 2017; Ronneberger et al., 2015; Tan & Le, 2019). Supervised segmentation approaches require abundant labeled images to achieve sufficient performance. Semi-supervised approaches provide a good approximation to supervised approaches. Chen et al., (2021) present an approach that imposes consistency between two networks and utilizes the pseudo segmentation map of unlabeled data to improve performance. Ouali et al., (2020) also prove the importance of consistency training for semi-supervised segmentation. Segmentation in unsupervised settings usually require post processing techniques and rarely achieve performance equivalent to supervised segmentation procedures. Xia et al., (2017) was the first to introduce unsupervised segmentation with deep learning. They use stacked U-Net networks as an encoder-decoder architecture. The k-way encoded image obtained from encoder is post processed using conditional random field, followed by hierarchical merging of segments. Recent studies make use of clustering to learn compact set of features (Cho et al., 2021; Ji et al., 2019).

Studies which propose active learning specifically developed for semantic segmentation are relatively new. Siddiqui et al., (2020) proposed viewAL that imposes consistency over multi view images. Lin et al., (2020) propose a new acquisition function based on segment entropy. They argue that pixels within the same segment must have the same network output. They use an unsupervised segmentation (Vosselman et al., 2017) procedure for computing the segments. Xie et al., (2021) incorporate a probability attention module that computes semantic difficulty of classifying pixels in an image. A difficulty aware entropy is used as the final acquisition function. Wu et al., (2022) combine informativeness with representativeness to acquire sub-regions rather than whole scenes. Although these methods achieve good performance, they have specialised applications.

The original formulation of active learning aimed at reducing the efforts required for data collection and annotation. As it is not trivial to estimate the efforts spent on data assimilation, most studies have used the number of annotations as a surrogate measure. This is not a correct metric in case of semantic segmentation and remote sensing applications where the cost of annotation might not be identical for all samples. To address this challenge. cost sensitive active learning methods have been developed (Demir et al., 2014; Geiß et al., 2018; Persello et al., 2014). However, all these methods try to optimize ground surveys which are deprecated due to the advancement in online annotation technology. Only one study, (Mackowiak et al., 2019) has incorporated the cost of online annotations into their active learning framework for semantic segmentation. In this study, the authors train a cost prediction module using the number of clicks required to annotate an image. They combine the estimated cost map with uncertainty map to form the final acquisition function. They experiment with three different arithmetic combinations of the cost and the uncertainty map. The major drawback of the approach is that they assume that the cost of annotations are given for an application.

3.2.2 Crop Classification

In this study, we focus on the problem of active learning for crop classification. Crop classification is an important task in remote sensing. Classified information about crops can be used for implementing agricultural policies, maintaining food security, and better crop management. Many regions at the global scale are food insecure where computing agricultural production is difficult due to their complex stratification. Crop maps are a good tool for quantifying agricultural production thus, accurate crop maps are required both at regional and global scales (Waldner et al., 2016). Such maps can also be used in a pipeline for other related tasks such as yield estimation, crop disease prediction, lead/cadmium contamination prediction in crops, etc (Ibrahim et al., 2021). Furthermore, annual crop maps may reveal changes in the local ecosystem, effects of climate change, or agricultural practices.

Previously, satellite image time series (SITS) have been used crop-type classification (Weikmann et al., 2021). Vuolo et al., (2018) has described the usefulness of multi-temporal images over mono-temporal image for crop-type classification. Nowakowski et al., (2021) has used transfer learning to reduce the amount of training samples required for crop classification. Z. Zhang et al., (2019) has presented preliminary results of a multi-view active learning and Niazmardi et al., (2019) has used multi-domain active learning for crop mapping. Brandt, (2019) has achieved state-of-the-art performance on time series crop mapping using capsule networks but uses a lot of training samples. Finally, Tseng et al., (2021) has presented an model agnostic meta learning (MAML) algorithm which make use of label abundant spaces to improve performance on sparce label spaces.

3.3 ALGORITHM

3.3.1 Notation and Setting

We consider pool based batch mode active learning setting. We follow the same procedure and notations as presented in 2.3.1 and 2.3.2.

3.3.2 Active Learning by Uncertain Class Diversity Conditioned on Target Site

Active Learning by Uncertain Class Diversity Conditioned on Target Site (UCD) presented in Algorithm 3 starts by pre-training the deep learning network on a randomly selected initial seed of patches from the training site. We compute the proportional distribution of classes in each training patch using the argmax of network predictions. If x_j^k is the j^{th} pixel in the k^{th} training patch then -

$$z_{ij}^k = \begin{cases} 1, & \text{if } i = \underset{c}{\text{argmax}}\, p_\theta(y = c|\, x_j^k) \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

$$p_d^k(c) = \frac{\sum_{j:c=i} z_{ij}^k}{|j|} \tag{3.2}$$

where we iterate over all $i \in \{1, 2, ...K\}$ and $j \in \{1, 2, ...n\}$. Here, K is the number of all possible classes and n is the number of pixels present in each patch. If a class (c^* say) is not present in the

Algorithm 3 Active Learning by Uncertain Class Diversity

Require: L, U, Q, S, C, k
Perform Pre-Train C on L
while not **done**:
    #Compute $p^k_{tr}(c)$ using 3.4
    #Compute $p_{te}(c)$ using 3.5
    $L^* \leftarrow \{x^{i_1}, x^{i_2}, ...x^{i_k}\} \leftarrow \underset{i \subset \{1,2,...\}, |i|=k}{\operatorname{argmin}} KL(p^i_{tr}||p_{te})$
    Label L^* using S
    $L \leftarrow L \cup L^*$
    Train C on L
end while

k^{th} patch, then we set $p^k_d(c^*) = 0$.

Then we compute the average uncertainty of each class present in the argmax of network predictions in each training patch. We use monte carlo dropout to model uncertainty and margin score to quantify it. This is given by -

$$p^k_u(c) = \frac{\sum_{j:c=i} z^k_{ij}(p_\theta(y_1|\, x^k_j) - p_\theta(y_2|\, x^k_j))}{\sum_j (p_\theta(y_1|\, x^k_j) - p_\theta(y_2|\, x^k_j))} \tag{3.3}$$

where y_1 and y_2 are the two largest class probabilities in the corresponding network prediction. Finally, we combine the uncertainty of classes with their proportional distribution -

$$p^k_{tr}(c) = \frac{p^k_d(c) p^k_u(c)}{\sum_i p^k_d(i) p^k_u(i)} \tag{3.4}$$

Similarly, $p^k_{te}(c)$ can be calculated for the k^{th} patch in testing site. Again, we set $p^k_{te}(c^*) = 0$ if a class c^* is not present in the k^{th} patch. The marginal distribution of each class conditioned over patches can be calculated by averaging over the uncertain class distribution (UCD) of all testing patches -

$$p_{te}(c) = \frac{\sum_k p^k_{te}(c)}{|k|} \tag{3.5}$$

where $k \in \{1, 2, ...P\}$ and P is the number of patches in testing site. Now we define the active learning problem as -

$$x = \underset{k}{\operatorname{argmin}}\, KL(p^k_{tr}||p_{te}) \tag{3.6}$$

This means that we select the training patch with the lowest KL divergence between its UCD and the marginal UCD of the testing patches. In batch mode, we acquire the top k training patches

with the lowest corresponding KL divergences -

$$\{x^{i_1}, x^{i_2}, ...x^{i_k}\} = \underset{i \subset \{1,2,...\}, |i|=k}{\text{argmin}} \; KL(p^i_{tr} || p_{te}) \tag{3.7}$$

3.4 STUDY AREA AND DATASETS

The study areas selected for this research are located in the United States of America (US) (figure 3.1). The training area (TA) considered for the study covers parts of Missouri, Kentucky and Tennessee. Further, we consider two testing sites. One testing site is located in Louisiana (TS1). The other testing site is located in Iowa (TS2). We use the United States Department of Agriculture (USDA) Crop Data Layer (CDL) as our reference dataset. USDA provides yearly CDL data starting from the year 1997, which covers the whole US. The CDL is a raster layer containing crop specific land use and land cover categories. Out of the 256 CDL categories, we focus on the four most common crops found in the study areas. These are corn, cotton, rice and soybean. Rest of the categories are merged into class 'other'.

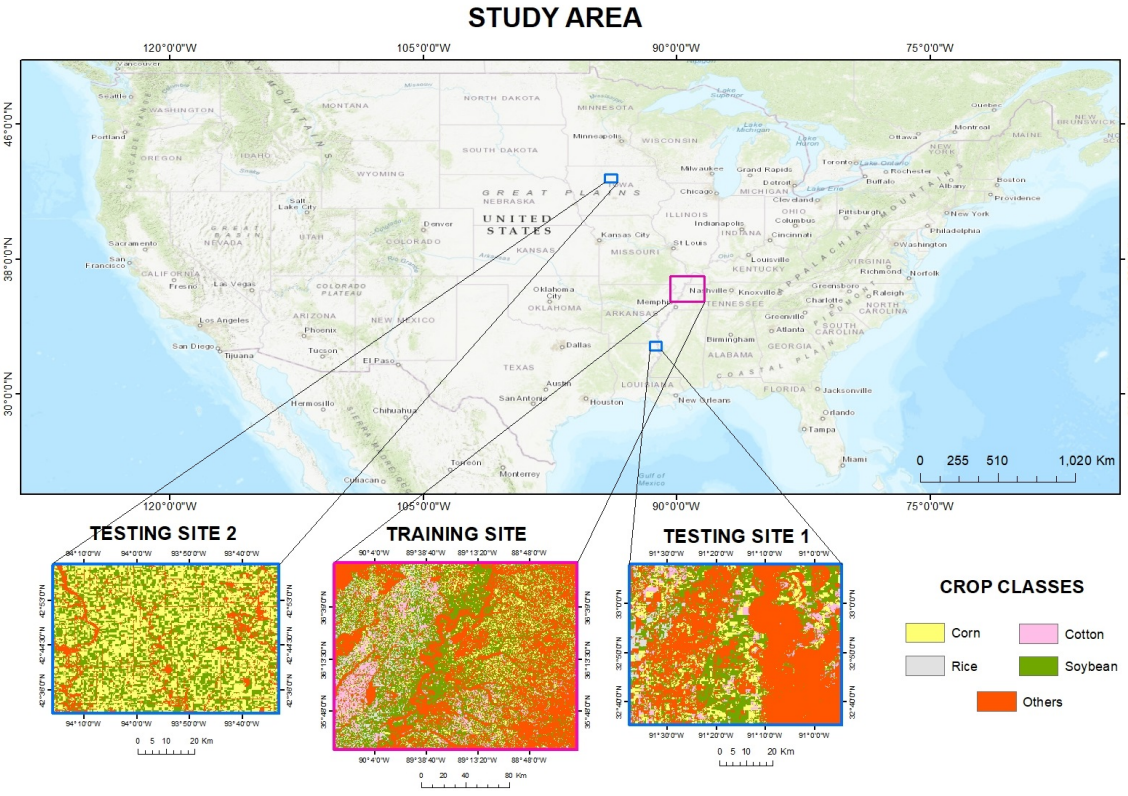


Figure 3.1: Location of the study areas in the United States of America.

For the classification, we use the Sentinel-2 L2A product. We preferred Sentinel-2 over Landsat imagery due to its superior temporal resolution. We compute monthly max NDVI composite using the Sentinel-2 imagery of year 2019. This is done to minimize the effects of clouds. Further, we retain a total of 8 images for months April to November. All the images are stacked and resampled to 30m resolution using bilinear interpolation (to match the spatial resolution of CDL). Some descriptions of the study areas and datasets are presented in Table 3.1 and Table 3.2.

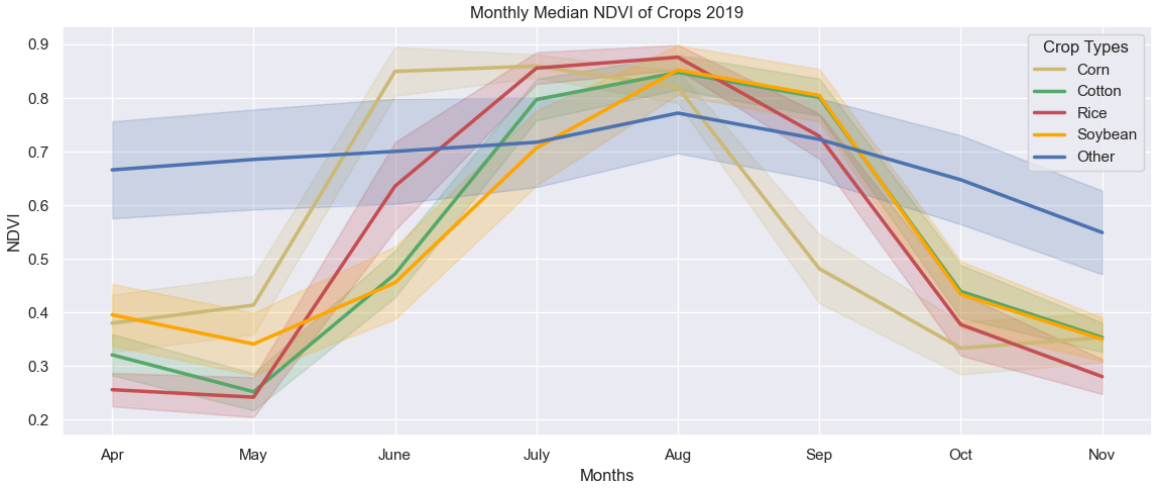


Figure 3.2: Temporal profiles of NDVI of the five classes considered for classification.

Table 3.1 Description of the datasets in consideration.

Description	Sentinel-2 L2A	Crop Data Layer
Temporal Resolution	5 days	1 year
Spatial Resolution	10m	30m
Radiometric Resolution	16 bits	8 bits
Source	ESA	USDA

Table 3.2 Descriptive statistics of the study areas in consideration.

Statistic	TA	TS1	TS2
Total Area (sq. km)	37436.87	5687.28	5687.28
Min NDVI	-0.34	-0.48	0.00
Max NDVI	0.99	0.99	0.99
Area Corn (sq. km)	4695.72	642.41	2810.96
Area Cotton (sq. km)	3265.50	258.21	0
Area Rice (sq. km)	1329.30	105.22	0
Area Soybean (sq. km)	11838.60	2694.14	1960.80

3.5 EXPERIMENTS

3.5.1 Experimental Setup

The NDVI SITS and CDL of TA, TS1 and TS2 are divided into patches of size 128x128 pixels. This is done to avoid memory overflows. TA contains a total of 2280 patches while TS1 and TS2 contain 300 patches each. We use the SegNet architecture (Badrinarayanan et al., 2015) for semantic segmentation of the NDVI SITS. As SegNet is a fully convolutional network, dropout cannot

Table 3.3 Performance metrics of SegNet trained on TA and tested on TS1 and TS2.

METRIC	TA	TS1	TS2
LOSS	31.81	41.49	48.56
ACCURACY	84.07%	79.34%	45.65%
MIOU	0.64	0.41	0.27
F1 SCORE	0.84	0.79	0.46

be used directly (as it is required for computing the posterior). Instead, we introduce dropblock (Ghiasi et al., 2018) layer (p=0.25) after each convolutional layer except the final convolutional layer which is used to generate the class probability maps.

We first train the network on all the available patches in TA (*epochs=200, Adam, lr=0.00007, batch=128*). We use early stopping and model checkpointing within the training. Randomely selected patches from TS1 and TS2 are used for validation during the training process. We experiment with four different loss functions - cross entropy loss, weighted cross entropy loss, focal loss and jaccard distance (see E). Jaccard distance produces the most stable training and validation performances. As it directly optimizes the mean Intersection over Union (mIoU), it produces the best classification results. We consider this as the baseline performance. The training results are present in Table 3.3

We notice that SegNet generalizes better on TS1 than TS2. This is because both TA and TS1 lie in the same Koppen climatic zone - Cfa (humid subtropical). TS2 lies in the climatic zone Dfa (hot-summer humid continental). This results in differences in crop management and cropping seasons between TA and TS2. Further, TS2 contains only two crop classes - corn and soybean. On inspecting the results of SegNet on TS2, we see that it produces majority of false positives of cotton over corn. This indicates that the NDVI temporal profile of cotton in TA is similar to that of corn in TS2.

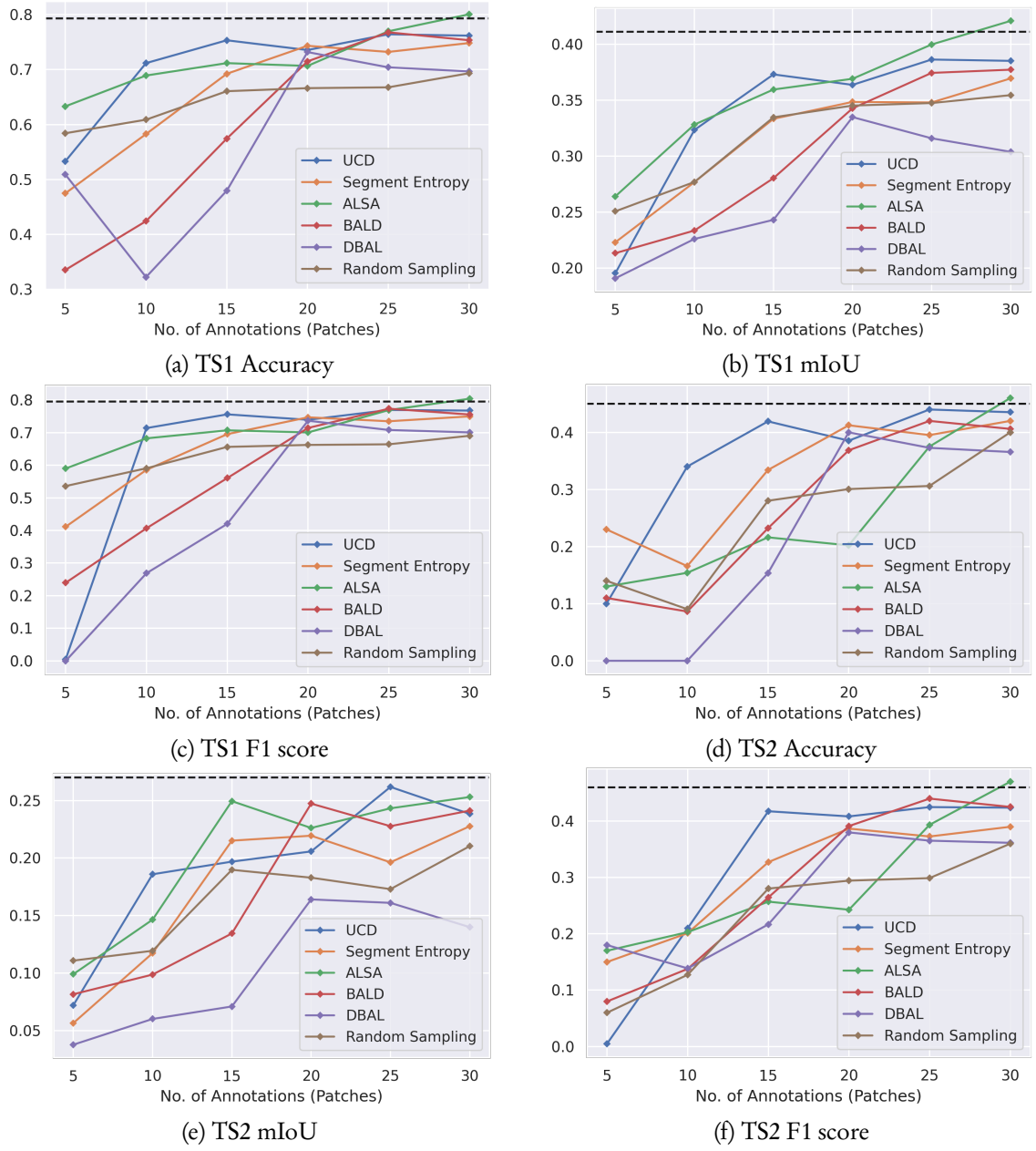
3.5.2 Results and Discussion

We compare the performance our proposed method with the following baselines on TS1 and TS2 -

1. **ALSA**: We use the convolutional variational autoencoder (Kingma & Welling, 2014) to embed the training patches to a latent space with dimension 500 (see F). These transformed vectors are used in the clustering step. We compute the margin score of a patch by averaging the margin score of each pixel within that patch. Rest of the algorithm is the same as described in Algorithm 2.

2. **Segment Entropy** (Y. Lin et al., 2020): We use felzenswalb's segmentation method (Felzenszwalb & Huttenlocher, 2004) to compute unsupervised segments within each patch. We then compute the entropy of network predictions within each segment and average it for all segments within a patch. We select the top k patches with the highest segment entropy.

3. **BALD**: We select the top k patches from the TA which have the highest mutual information



(a) TS1 Accuracy (b) TS1 mIoU

(c) TS1 F1 score (d) TS2 Accuracy

(e) TS2 mIoU (f) TS2 F1 score

Figure 3.3: Classification performance (on testing sites) of various active learning algorithms as a function of number of annotated images using SegNet.

among the classifiers as described in (Houlsby et al., 2011). This score is calculated for each patch by averaging each pixels' score. We use 10 MC samples to model uncertainty.

4. **DBAL**: We select the top k patches from the TA which have the highest entropy. This score is calculate for each patch by averaging the each pixels' score. We use 10 MC samples to model uncertainty.

5. **Random Sampling**: We randomly select k patches from the TA.

Again, we report the results by taking an average over 3 runs (figure 3.3). Dropout layers are still

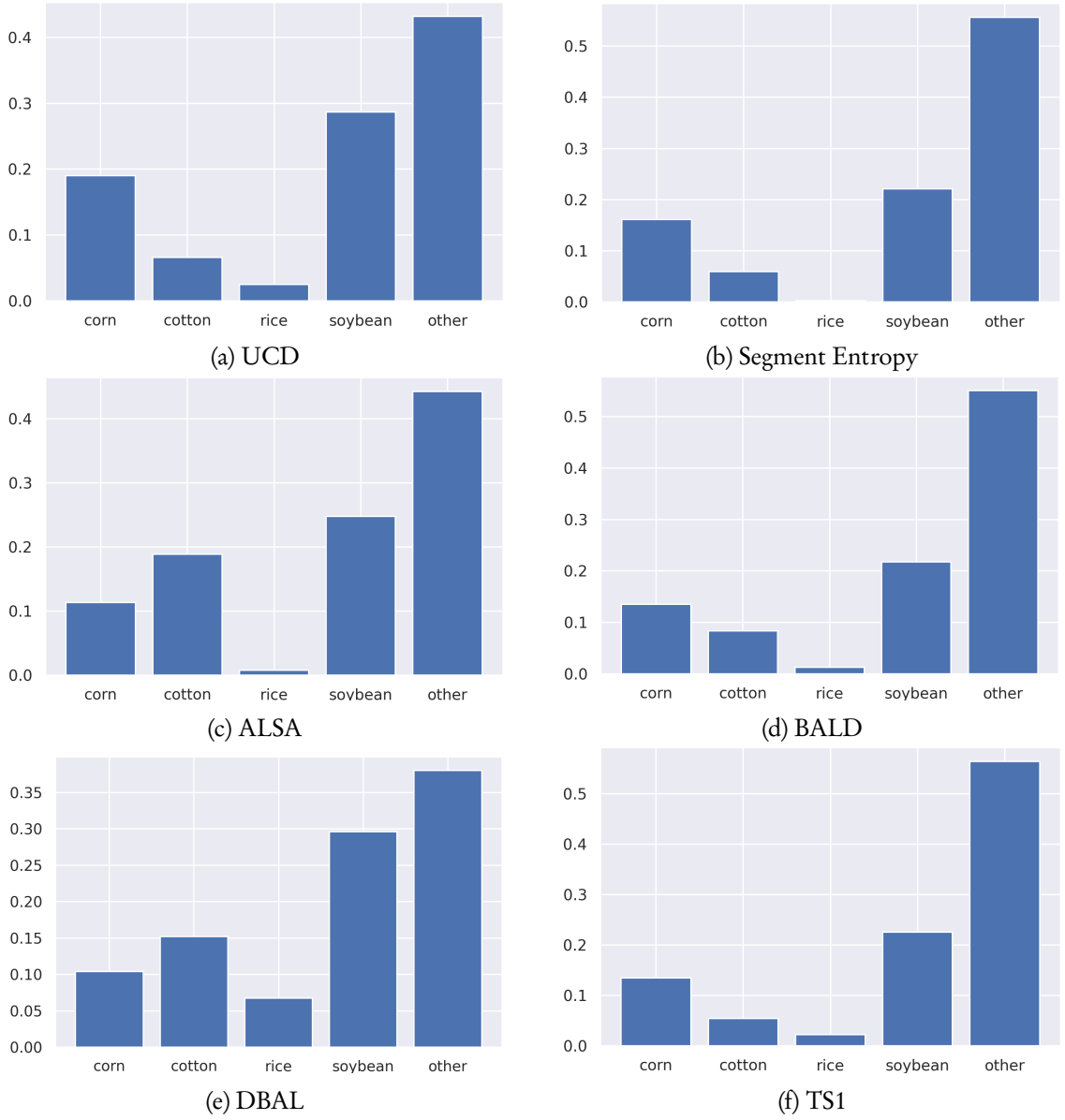
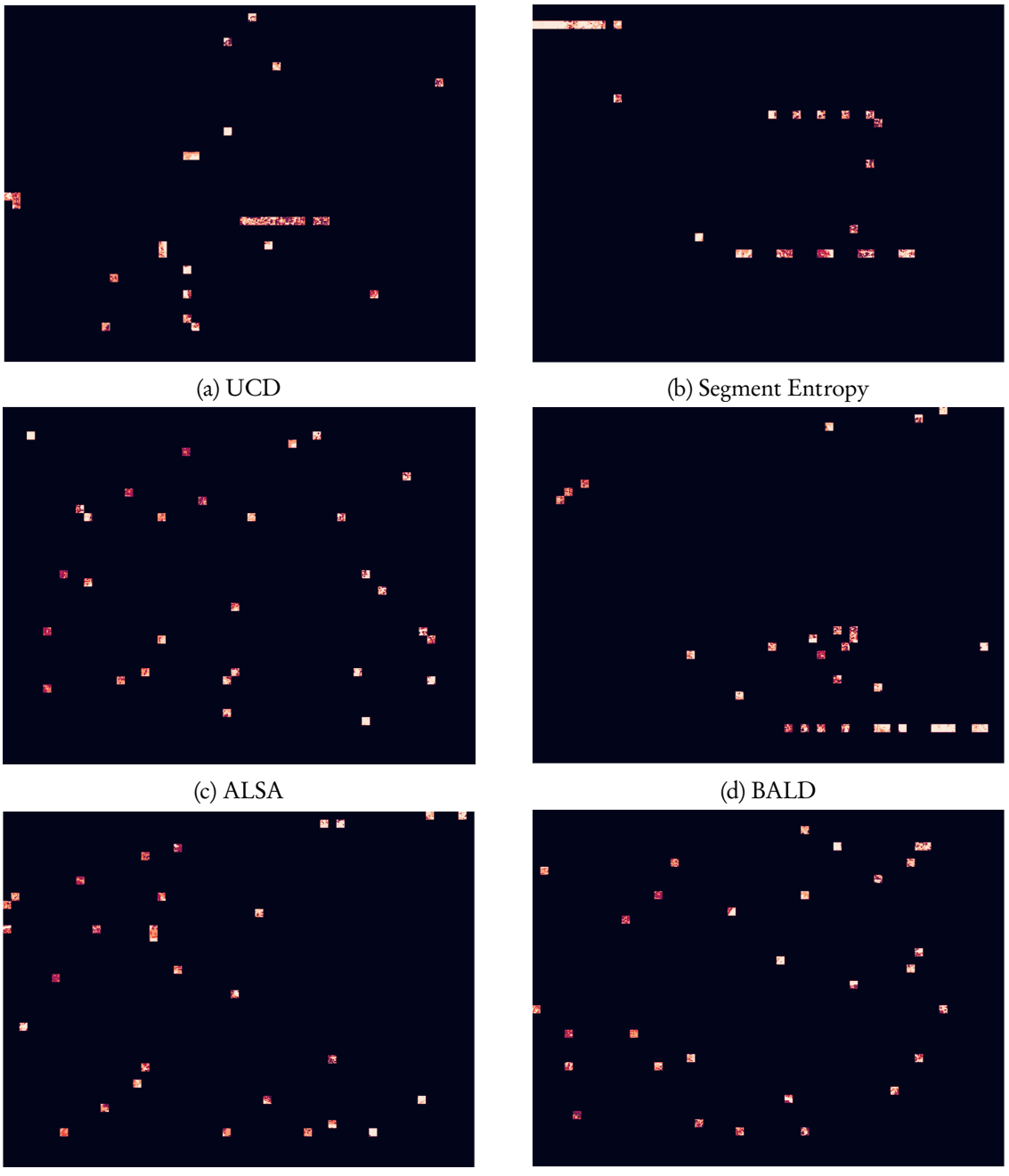


Figure 3.4: Class distribution of final training set resulting from the active learning methods.

included for non-BNN active learning methods. We fix the acquisition batch size to 5 patches per iteration. Firstly, we notice that the performance of the active learning methods is not as stable as it was in the image classification experiments. This raises the question of feasibility of active learning methods for semantic segmentation. However, they tend to converge to the baseline performance very quickly. ALSA is able to outperform all other active learning methods. This is possibly due to the cluster sampling step as rest of the active learning methods are purely uncertainty based. However, ALSA has the worst runtime compared to the rest of the methods due to its embedding and clustering step. This makes its application unsuitable for practical real world problems.

UCD, on the other hand, has the second best performance. It tries to match the class distribution of data in the training and testing area. Further, it has a feasible runtime making it appropriate for practical applications. Segment entropy performs better than the traditional uncertainty based



(a) UCD (b) Segment Entropy

(c) ALSA (d) BALD

(e) DBAL (f) Random

Figure 3.5: Spatial distribution of the patches acquired by the active learning methods.

methods - BALD and DBAL. However, its performance and efficiency is highly dependent on the quality of segmentation attained. BALD and DBAL barely outperform random sampling in the final set of acquisitions. The prime reason for such a result is their biased evaluation of uncertainty. A single patch may contain multiple classes and averaging over pixel uncertainty might not account for the class diversity present in the patch.

Figure 3.4 shows the distribution of classes in the final labeled dataset generated by the active learning methods. The class distribution is measured by the ratio of pixels present in each class in

the labeled dataset. Though BALD and Segment Entropy produce similar class distribution as in TS1, they do not perform equally well as ALSA or UCD. This is because they compute patchwise uncertainty instead of classwise uncertainty. The cluster sampling step of ALSA is able to account for uncertainty over clusters, which is indirectly representative of the classes.

Figure 3.5 shows the location of patches acquired by different active learning methods. This figure presents an interesting result that good spatial coverage of training samples does not necessarily result in good classification results. This is the primary difference between active learning and sampling techniques. Spatial sampling techniques try to optimize the spatial distribution of samples while completely ignoring the expected performance of models they will serve as inputs in. Visually, both DBAL and random sampling tend to acquire samples all over TA. However, both these methods are the worst performing. On the other hand, BALD, segment entropy and UCD acquire samples from specific sites in TA. The patches acquired by ALSA are scattered due to the cluster sampling step. This shows that quality of patches is equally important as compared to spatial coverage.

ALSA and UCD are able to achieve close to baseline performance with just 30 patches in the training set. This is equivalent to utilizing about 1.31% of the available unlabeled TA. When we translate this result to the ground reality, it implies that only 492.59 km^2 of TA is annotated out the available 37436.87 km^2. Additionally, it indicates that upto 75x efforts can be reduced that is generally spent on collecting and annotating data with these active learning methods. The time and capital saved on data assimilation in one region can be spent on other regions where data is not available.

3.6 CONCLUSIONS

In this chapter, we presented an active learning method for semantic segmentation that models pseudo class wise uncertainty as opposed to sample wise uncertainty used by other methods. This allows our method to take advantage of the class distribution present in the testing dataset. Though ALSA performs the best in our experiments, its computational overhead makes it unsuitable for applications in semantic segmentation. Further, we presented evidence that quality of training samples is as important of a driving factor as spatial coverage in semantic segmentation. In the future, the proposed acquisition function may be combined with a diversity measure to account for intra class variance. When several testing sites are present, the framework maybe be adapted to alternatively optimize acquisition for the testing sites.

Chapter 4

Active Meta Learning for Few Shot Learning

> Persons with comparatively moderate powers will accomplish much if they apply themselves wholly and indefatigably to one thing at a time.
>
> Samuel Smiles in *Self-help*, 1866

4.1 INTRODUCTION

Up until now, we presented novel active learning algorithms for offline learning. In offline learning, the newly acquired data points are combined with the existing labeled dataset. The network weights are reset and trained from scratch. This procedure happens in every iteration of the active learning loop. The limitation of this approach is that the computational time increases as the size of the labeled dataset grows. Further, we argue that an oracle is only able to annotate a limited set of examples at any given time. For example, it took 1.5h on average to annotate a single image of the Cityscapes dataset (Cordts et al., 2016) at fine level. If offline learning is used in such cases, then either the network will remain untrained for a long time or lead to poor generalizations if trained with very few data points. As a result, online active learning is a better alternative in applications with continuous streaming data.

Few shot learning (FSL) is a common setup in online learning, where network has to learn from a few given examples at each time step. FSL has seen applications in character generation, robotics, drug discovery and so on (Y. Wang et al., 2020). In remote sensing, FSL has largely been used for scene classification (Alajaji et al., 2020). Given the definition, FSL was initially developed to address classification problems. However, recent studies have utilized its potential to also solve regression problems. FSL was built on the foundation that traditional training procedures shall lead to a poor generalization of networks when working with insufficient data. A FSL problem is generally defined as a N-way k-shot classification problem. Here, the classifier has to discriminate between N distinct classes given only k examples per class. In practical setups, the value of N is much larger than the value of k. When k=0, FSL is known as zero shot learning and when k=1, FSL is called one shot learning.

There are various techniques that can be used to address a FSL problem. These are described in the section 1.1. In this section, we describe the meta learning problem which is relevant to our work. The aim of meta learning is to learn a distribution of tasks and then be able to generalize

well on unseen tasks. Meta learning achieves this by learning an average set of parameters over the given tasks. Each of the tasks translate to a FSL problem and is associated with a separate dataset. There are two main stages of meta learning training procedure - 1) meta train, where classifier is trained on a small support set; 2) meta update, where classifier is provided with an extra set of examples to compute the average set of parameters over the tasks. The extra examples form the query set for the meta learner.

In remote sensing, meta learning can be used for better spatial transferabilty of networks by learning the average parameters of different spatial tasks. Similarly, it can also be used for achieving temporal transferability of networks. All these problems fall under the category of semantic segmentation. However, meta learning is largely defined for image classification tasks and it is still unclear how this definition can be adapted for semantic segmentation tasks. This is obviously challenging as semantic labelling of images shall require us to amend the notion of N-way k-shot. How do we control the number of classes present in the FSL dataset of a semantic segmentation task? Do we consider a single image or a single pixel as a data point? Does the size of the images affect the performance of meta learning?

To the best of our knowledge, there exists no clear answers to these questions. Further, in the meta learning setup, the meta learner is expected to be provided with sufficient number of tasks. However, this is not always the case when it comes to applications where the collection of unlabeled data is itself expensive and time consuming. Even if unlabeled data is made available, labeling that is extremely time consuming. As a result, meta learning can only be applied in certain practical applications, where a large number of tasks are available. Meta learning is easily prone to overfitting when operating with overparametrized models and insufficient tasks. Further, the tasks designed for meta learning need to be mutually exclusive for it to generalize over unseen tasks. Yin et al., (2019) present the problem of function-level overfitting of meta learning when trained on mutually inclusive tasks. To address these challenges, active learning may be used to select appropriate data points for faster and better convergence.

In this study, we propose to use active learning that selects optimal tasks for meta learning. Once the task is selected, active learning is used to further select data points within that task. Specifically, we use model agnostic meta learning (MAML) for the experiment. For the active task selection, we use the BALD score over the data points in each task. We present the results of one shot learning with and without MAML. Further, we compare the results of random sampling of tasks against the active meta learning framework. With active meta learning, we train only on the newly added data points in each iteration. This allows a lower cost and faster operations. The contributions of the work are twofold -

1. We use active learning for task selection and data annotation within a meta learning framework for semantic segmentation.
2. We present a comparison of online active learning setup with meta learning and vanilla gradient descent.

The rest of the chapter is organised in four sections. The next section presents a survey of meta learning methods. After discussing some of the related works, we present our proposed method - *Active Meta Learning for Spatial Transferability*. The following section describes the experimental results achieved on the task of one shot crop classification. Finally, we present some limitations and possible future works in this direction.

4.2 RELATED WORK

Meta learning methods can be categorized according to their overall objectives (Hospedales et al., 2021). These broad categories are namely metric based, model based and optimization based. These methods are briefly described below -

Metric based. The idea of metric based learning is to model the distance between a given set of data points and then classify them based on this distance measure using an algorithm such as the k-nearest neighbour. The neighbours are usually weighted using a kernel function that measures the similarity between data points. Siamese network (van der Spoel et al., 2015) uses two parallel convolutional neural networks to compute the embedding of two given images. These embeddings are used to predict whether the images belong to the same class. Matching network (Vinyals et al., 2016) uses an attention mechanism to predict whether an unseen observation belongs to previously seen support set. Prototypical network (Snell et al., 2017) encodes given images into a latent space using an embedding function. This space is used for classification. Relation network (Sung et al., 2018) is composed of an embedding module and a relation module. Unlike siamese network, it outputs a similarity score instead of a hard binary classification value.

Model based. These approaches try to utilize the power of neural networks to learn faster. Memory augmented neural networks (MANN) utilize external memory buffers to efficiently assimilate new and unseen data. Santoro et al., (2016) present MANN with a neural turing machine as backbone for meta learning. Meta Network (Munkhdalai & Yu, 2017) proposes a technique to optimize the weights of a neural network faster than stochastic gradient descent (SGD). They combine slow weights from SGD and fast weights from a network prediction to compute the final set of weights.

Optimization based. These approaches try to modify the learning procedure of neural networks for achieving better generalization with small number of data points. The general framework of optimization based meta learning methods is depicted in figure 4.1. MAML (Finn et al., 2017) proposes two step gradient descent procedure. First step computes task specific parameters over randomely sampled set of tasks. The next step updates the neural network with an average set of parameters computed over a query set. Yoon et al., (2018) proposed bayesian MAML using stein variational stochastic gradient descent (SVGD) that approximates task posterior for better and faster convergence. They use a novel chaser meta-loss that minimizes the distance between predicted task posterior and true task posterior. Antoniou et al., (2019) proposed simple modifications in the training procedure of MAML for better stability and faster convergence. Specifically, they solved the problem of gradient instability, high second order derivative cost, batch normalization bias and fixed learning rate. Ravi et al., (2017) proposed a LSTM based meta learner to model the learning procedure of another neural network that is used for few shot learning.

To the best of our knowledge, only three studies (Ruswurm et al., 2020), (Tseng et al., 2021) and (Tseng et al., 2022) have evaluated MAML for spatial transferability. They compare the results of MAML-trained and pre-trained models. Ruswurm et al., (2020) experimented on the DeepGlobe dataset for land cover segmentation. They reported that MAML outperforms regular gradient descent when the train and test domains are diverse. In case the train and test domains are similar, MAML fails to perform well. This is mostly due to the lack of enough data. Tseng et al., (2021, 2022) performed one shot binary crop classification. They noticed that MAML is robust to class imbalances but highly sensitive to overfitting when there is a lack of data for validation. This is yet another evidence to support our hypothesis that meta learning with insufficient data could perform otherwise.

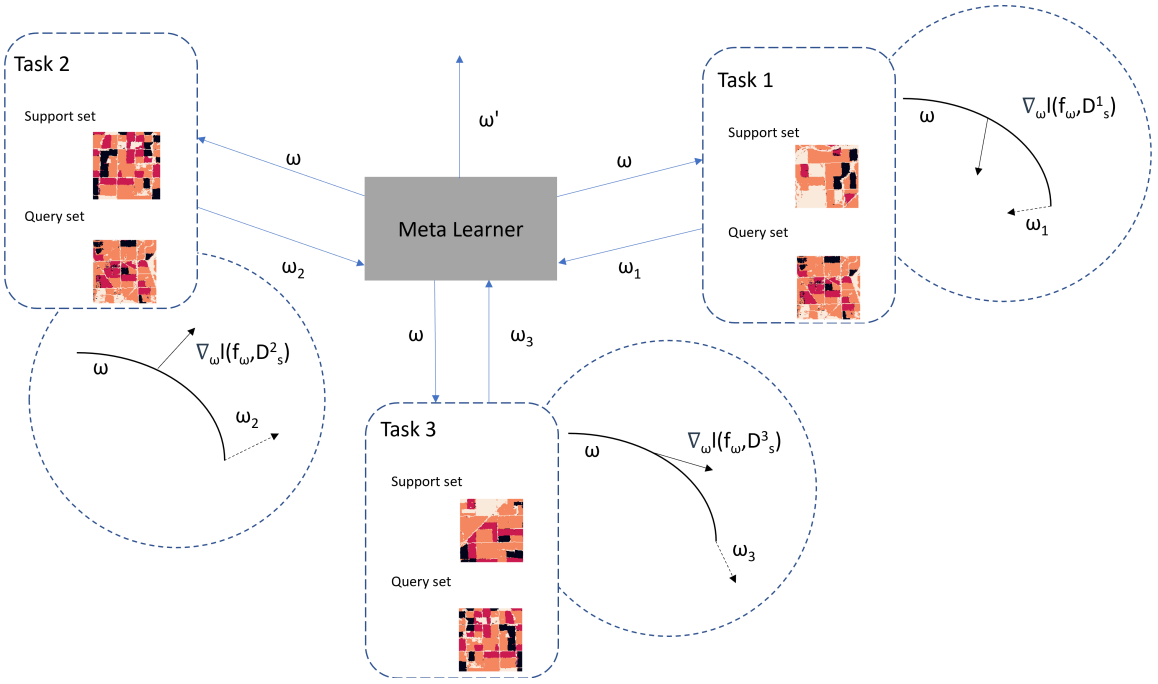


Figure 4.1: Meta learning using optimization based parameter refining technique.

To address the issue of insufficient tasks in meta learning, a generalized regularization based meta learning methods have been proposed in the past (Balaji et al., 2018; Jamal et al., 2018). Yet other methods try to augment tasks using sophisticated techniques (Lee et al., 2019; Yao et al., 2021). They do not take into consideration the sequential nature of data collection and annotation process in practice. Not all tasks are equally important for generalization of meta learner over unseen tasks. Further, random sampling of tasks may lead to sub optimal results (Mehta et al., 2019). Hence, active learning can used to optimally design episodes for meta learning. Kaddour et al., (2020) proposed probabilistic active meta learning approach to select the next optimal training task for meta learner. They make use of task descriptors to search over a given task and use the information for selecting the next task.

4.3 ALGORITHM

4.3.1 Notation and Setting

Meta learning consists of a distribution of tasks $\tau \sim T = \{\tau^1, \tau^2..., \tau^n\}$. Each task τ is associated with a support set D_s and a query set D_q. Both datasets contain data points in the form of image-label pair. An image is of the form $x^i \in \mathbb{R}^{nxmxc}$ and the corresponding labels are of the form $y^i \in \mathbb{N}^{nxm}$. In case of image classification, D_s is of the form N-way k-shot. However, in case of semantic segmentation, there is no control over the distribution of classes (in terms of the number of pixels of each class) present in a single image. As a result, we have to define few shot learning problem by the number of images present in D_s. We denote the segmenting network as f_ω and the loss function as $l(f_\omega, x^i)$.

4.3.2 Model Agnostic Meta Learning

MAML, presented in Algorithm 4, starts by randomely initializing the weights of the segmenting network in use. The algorithm has access to a bunch of tasks T. In an iterative fashion, it randomely samples a batch of tasks. For each task, it calculates the task specific parameters by optimizing the loss over its support set. This is equivalent to having task specific networks f_{ω_i}. The outer loop of MAML updates the original network parameters using the gradient of loss calculated with f_{ω_i} networks over the corresponding query sets. Note that MAML requires separate learning rates for the inner and outer loops.

Algorithm 4 Model Agnostic Meta Learning

Require: T, f_ω, learning rates - α, β
Perform Randomly initialize ω
while not **done**:
  Sample $\{\tau^1, \tau^2..., \tau^n\} \sim p(T)$
  for each $\{\tau^1, \tau^2..., \tau^n\}$:
    Sample $D_s = \{x^j, y^j\} \sim \tau_i$ of size k points
    Sample $D_q = \{x^j, y^j\} \sim \tau_i$
    $\omega_i \leftarrow \omega - \alpha \nabla_\omega l(f_\omega, D_s)$
  end for
  $\omega \leftarrow \omega - \beta \sum_{\tau_i} \nabla_{\omega_i} l(f_{\omega_i}, D_q^{\tau_i})$
end while

4.3.3 Active Meta Learning for Spatial Transferability

The overall framework is presented in figure 1.1. We consider the problem of one shot crop classification. The first step of applying MAML for crop mapping is to define mutually exclusive tasks. We do this by dividing TA into ten parts of equal size (see figure 4.2). The division is made vertically, as climate changes across latitudes. We call them *spatial tasks*. Active meta learning, presented in Algorithm 5, replaces the task sampling step of MAML with an active selection algorithm. The active selection algorithm comprises of two stages - 1) selecting current best task; 2) sampling support and query set from the selected task in step 1. Both these steps involve a single iteration of separate active learning algorithms.

We propose to use a adapted BALD score for sampling the current best task. The task with the highest BALD score is selected for training the MAML in the current iteration. The traditional BALD score is calculated using the shannon entropy over model predictions and parameters. In the case of semantic segmentation, the shannon entropy of an image is given by averaging the shannon entropy score over all the pixels. If x_j^k is the j^{th} pixel in the k^{th} training patch, then the shannon entropy given by:

$$H(y^k|x^k, \omega) = \frac{\sum_j E_{p(y_j^k|x_j^k,\omega)}[-log(p(y_j^k|x_j^k, \omega))]}{|j|} \tag{4.1}$$

Algorithm 5 Active Meta Learning

Require: T, f_ω, Q, S, n, learning rates - α, β, γ
while not **done**:
  Set $T_s \leftarrow \phi$, $\omega^{'} \leftarrow \omega$
  for $i = 1$ to n **do**:
    $\tau_i \leftarrow \underset{\tau}{\text{argmax}} B(\tau, \omega^{'})$
    #Compute $p^k_{\tau_i}(c)$ using 3.4
    #Compute $p_{te}(c)$ using 3.5
    $\{x^{i_1}, x^{i_2}\} \leftarrow \underset{i \subset \{1,2\}, |i|=k}{\text{argmin}} KL(p^i_{\tau_i} || p_{te})$
    Label $\{x^{i_1}, x^{i_2}\}$ using S
    $D^{\tau_i}_s = \{x^{i_1}, y^{i_1}\}$
    $D^{\tau_i}_q = \{x^{i_2}, y^{i_2}\}$
    $\omega^{'} \leftarrow \omega^{'} - \gamma \nabla_{\omega^{'}} l(f_{\omega^{'}}, D^{\tau_i}_s)$
    $T_s \leftarrow T_s \cup \{D^{\tau_i}_s, D^{\tau_i}_q\}$
  end for
  for each $\{\tau^1, \tau^2 ..., \tau^n\} \in T_s$:
    $\omega_i \leftarrow \omega - \alpha \nabla_\omega l(f_\omega, D^{\tau_i}_s)$
  end for
  $\omega \leftarrow \omega - \beta \sum_{\tau_i} \nabla_{\omega_i} l(f_{\omega_i}, D^{\tau_i}_q)$
end while

The BALD score is then given by:

$$B(y^k, \omega | x^k) = H(y^k | x^k, \omega) - E_{p(\omega|x)}[H(y^k | x^k, \omega)] \tag{4.2}$$

To select the current best task, the challenge is to compute the BALD score over all the images in the task. We propose to evaluate the disagreement not only between the committee but also between the images in the task. We use the same principle as in BALD. The first term is computed as average entropy of the average predictive images generated from all the images in a task. We compute the average predictive image as -

$$\bar{y}^k = E_{p(\omega|x)}[p(y^k | x^k, \omega)] \tag{4.3}$$

The posterior $p(\omega|x)$ is estimated using MC dropouts. By applying dropout at the inference stage, we draw MC samples of the likelihood function of unlabeled samples. The second term is computed as the average entropy over the averaged entropy of all the predictive images in a task. The BALD expression used for acquisition of the tasks is presented below -

$$B(\tau_i, \omega) = \frac{\sum_k H(\bar{y}^k | x^k, \omega)}{|k|} - \frac{\sum_k E_{p(\omega|x)}[H(y^k | x^k, \omega)]}{|k|} \tag{4.4}$$

where $k \in \{1, 2, ...P\}$ and P is the number of patches in some task τ_i.

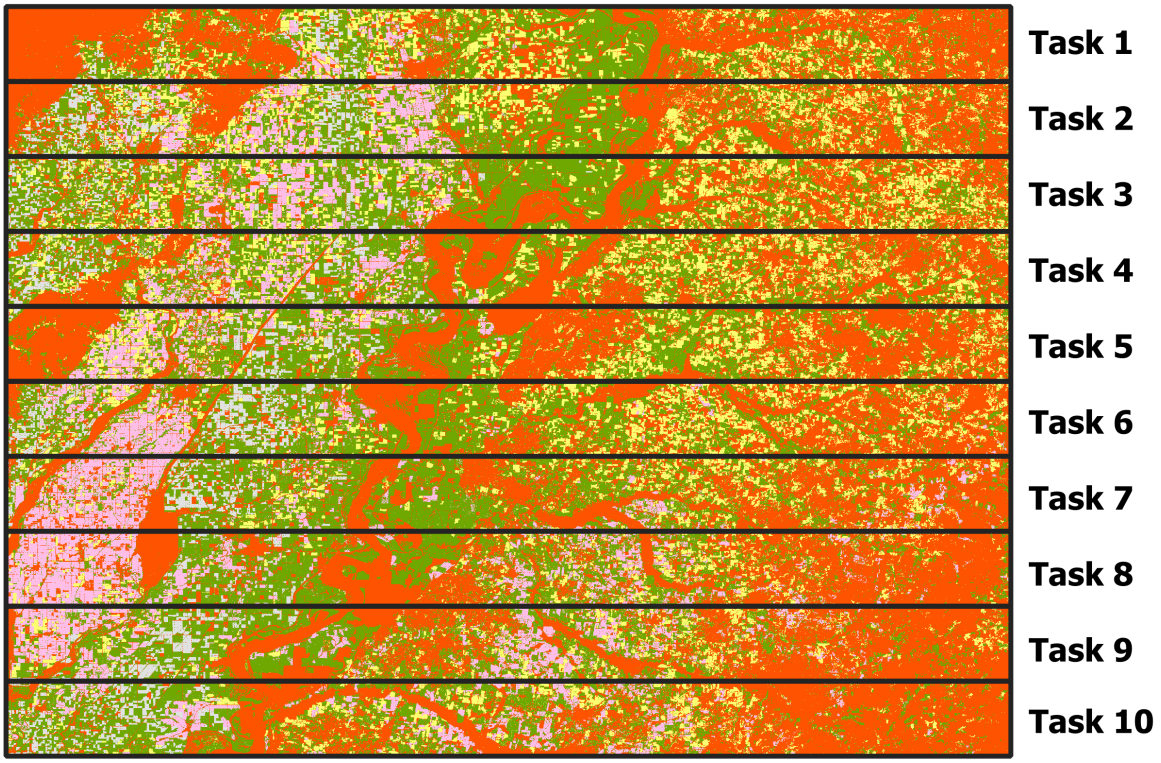


Figure 4.2: Design of spatial tasks for one shot crop classification.

After a task is selected, the next step is to sample a support and a query set. This is done using the active learning method proposed in Algorithm 3. After the support set D_s and query set D_q are sampled, we update the segmenting network with task specific parameters using D_s. This is done to ensure that the active learning algorithm samples distinct task in every iteration. However, if a task is already selected in the previous iteration, the next best task is selected. Before the beginning of the meta learning iteration, the segmenting network is updated with the original set of parameters as was before the active learning loop. The algorithm then follows the meta learning procedure presented in Algorithm 4.

4.4 EXPERIMENTS

We use a similar experimental setup as presented in 3.5.1. The training dataset contains patches of size 128x128 pixels. We create 10 arrays denoting the tasks and store these patches into their respective task array. The total number of patches present in TA is 2280 and each task array consists a total of 228 patches. The task arrays are loaded into the memory one at a time to avoid memory overflows. Further, we use the same segmenting network as was used in the previous chapter. For MAML, we use the SGD optimizer with inner and outer learning rates set to 1e-3. Please note that we experimented with various optimizers and found SGD to perform the best with MAML. For the active task selection loop, we use the Adam optimizer with a learning rate of 1e-4. We set the number of tasks per episode to 5. This means that we select 5 tasks per iteration of MAML. After each task is selected, we query two images from its task array using the active learning procedure. One image is placed in the support set while the other is placed in the query set. This is done

randomely. Further, we apply 5 gradient updates per task in the inner loop of MAML.

Table 4.1 Performance metrics of one shot crop classification using MAML trained on TA and tested on TS1 and TS2.

METRIC	TA	TS1	TS2
LOSS	39.21	45.62	51.16
ACCURACY	72.03%	55.59%	29.35%
MIOU	0.48	0.33	0.16
F1 SCORE	0.66	0.52	0.27

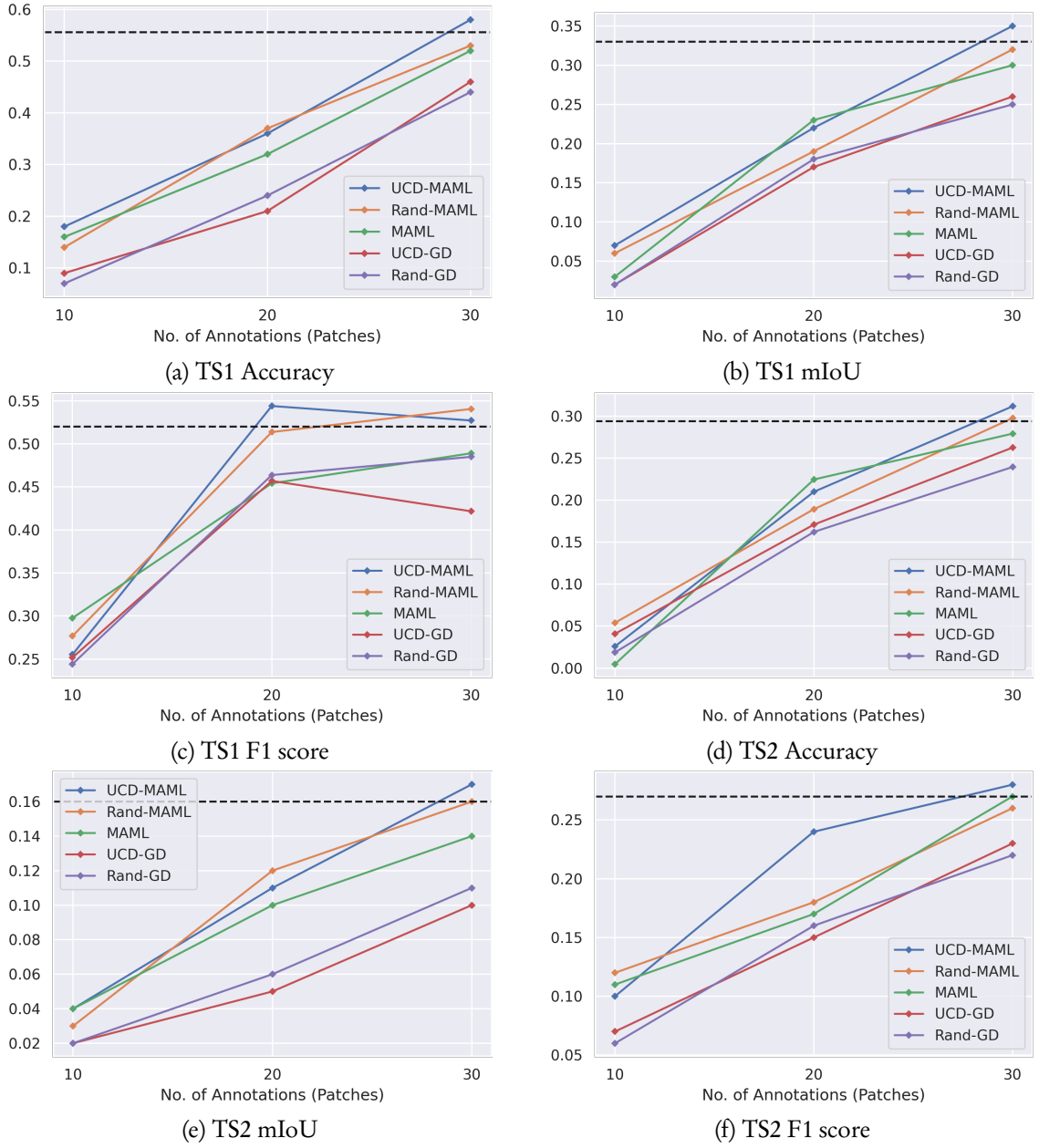
Using the same settings described above, we train MAML on all the available patches. This is considered as the baseline performance. We use random selection of tasks and the support and query set. We train the MAML exhaustively until it reaches convergence. The performance metrics are presented in table 4.1. We notice that there is a performance dip of at least 20% in case of one shot crop classification as compared to batch learning (see table 3.3). However, the training time of MAML is just 21 seconds as compared to 11.66 minutes of batch learning.

We now provide preliminary results of active meta learning on TS1 and TS2. We compare the results of the following methods -

1. **UCD-MAML**: This is our proposed method presented in Algorithm 5.
2. **Rand-MAML**: We select the tasks randomely while sample the support and query set using UCD. The training is performed using MAML.
3. **MAML**: We select the tasks randomely. The support and query set are also sampled randomely. The training is performed using MAML.
4. **UCD-GD**: We select the tasks using the active learning method proposed in Algorithm 5. We sample the support and query set using UCD. The training is performed using vanilla gradient descent.
5. **Rand-GD**: We randomly select the tasks, the support set and the query set. The training is performed using vanilla gradient descent.

The annotations of the support and query images are considered as separate. Further, we report the results of MAML based models after each episode is trained. Each episode contains a total of 10 annotated images. The vanilla gradient descent refers to training procedure where we train the network one task at a time (that is one image at a time). The support and query set are combined into a single dataset for this case. For training, we use model checkpointing and early stopping. To keep the evaluations similar, we report the results of vanilla gradient descent after it has been trained on 10 images.

As expected, our proposed method performs better than the rest of the methods used for comparison (see figure 4.3). In general, we see that the methods using MAML for training perform better than methods using vanilla gradient descent. This confirms the previous hypothesis that batch learning performs poorly when training in an online fashion. When using vanilla gradient



(a) TS1 Accuracy (b) TS1 mIoU

(c) TS1 F1 score (d) TS2 Accuracy

(e) TS2 mIoU (f) TS2 F1 score

Figure 4.3: Classification performance (on testing sites) of MAML and vanilla gradient descent based active learning methods for one shot crop classification.

descent for online learning, deep learning models tend to forget the knowledge acquired from previous tasks. This is also known as the problem of catastrophic forgetting in neural networks (J. Kirkpatrick et al., 2017). Data insufficiency further aggravates this problem in online learning. Recent studies have shown the effectiveness of meta learning for solving such problems in online and continual learning (Masarczyk & Tautkute, 2020; Yap et al., 2020).

Tseng et al., (2021) noticed that increasing the number of gradient steps in MAML can improve performance only when there is availability of enough data for validation. In our experiments, we set the number of gradient steps to 5 as the training with 10 gradient steps was highly unstable.

Ruswurm et al., (2020) in their work concluded that data inadequacy is a bottleneck for spatial transferability using MAML. Our results illustrate that active learning can effectively address this problem in MAML. In this experiment, the domains of TA and TS1 is similar while that of TA and TS2 is different. In both cases, the active meta learning method achieves performance similar to the baseline MAML model.

4.5 CONCLUSIONS

In this chapter, we presented a novel framework for sequential selection of tasks for few shot learning. Our framework was evaluated in an online learning setup for one shot crop classification. The inclusion of MAML for training improved the performance over vanilla gradient descent. However, MAML starts overfitting if the number of gradient steps are increased. The training time with online learning is much less as compared to offline learning. Hence, this kind of setup can be extremely useful in applications with continuous flow of data. In the future, the active learning algorithm may be adapted to select tasks in a batch mode. This might be beneficial in applications with the availability of several training tasks. This framework can also be evaluated with different patch sizes and neural network architectures.

Chapter 5

Conclusions and Recommendations

> You'll never reach perfection because there's always room for improvement. Yet get along the way to perfection, you'll learn to get better.
>
> Hlovate in *Versus*, 2010

5.1 PARETO ANALYSIS

Here, we provide the performance-input ratio of the active learning methods considered in the entire study. First, we present the ratio for active learning methods in Chapter 2. Then, we report the average ratio over the results achieved on TS1 and TS2 in Chapter 3 and Chapter 4. The performance ratio is calculated as performance at a given input size divided by the performance when entire input pool is used. The input ratio is calculated as the number of data points in labeled training set divided by the total number of data points available. Both the metrics are expressed as percentages.

Table 5.1 Performance ratio to input ratio of active learning methods considered in the Chapter 2.

Method	Cifar10	UC Merced	EuroSat
ALSA	80.95:20	103.53:20.3	92.32:3.16
Cluster Margin	77.83:20	91.76:20.3	92.40:3.16
BADGE	78.50:20	90.58:20.3	92.13:3.16
BALD	75.93:20	91.76:20.3	91.82:3.16
Core Set	80.13:20	96.46:20.3	93.50:3.16
DBAL	76.73:20	91.76:20.3	89.56:3.16
Random Sampling	76.49:20	76.46:20.3	89.78:3.16

We report that all the active learning methods including random sampling clearly achieve super pareto on *Mnist*, *Fashion Mnist* and *EuroSat*. However, on *Cifar10* (see table 5.1), only ALSA and Core Set are able to beat the pareto. While, the results on *UC Merced* illustrate that random sampling is unable to beat the pareto. Furthermore, in table 5.2 and table 5.3, we report that random sampling method is able to achieve super pareto. Overall, the results suggest that training with few good quality data points may result in sufficiently high performance. Adding more data points

Table 5.2 Performance ratio to input ratio of active learning methods considered in the Chapter 3.

Method	Accuracy	mIoU	F1 score
UCD	96.02:1.32	91.61:1.32	96.45:1.32
Segment Entropy	94.31:1.32	86.57:1.32	94.22:1.32
ALSA	100.90:1.32	103.08:1.32	101.03:1.32
BALD	94.95:1.32	89.07:1.32	94.93:1.32
DBAL	87.81:1.32	65.48:1.32	88.04:1.32
Random Sampling	87.38:1.32	81.78:1.32	86.75:1.32

Table 5.3 Performance ratio to input ratio of active learning methods considered in the Chapter 4.

Method	Accuracy	mIoU	F1 score
UCD-MAML	100.07:1.32	106.06:1.32	100.96:1.32
Rand-MAML	97.13:1.32	99.02:1.32	100.08:1.32
MAML	97.22:1.32	89.02:1.32	96.34:1.32
UCD-GD	84.81:1.32	74.25:1.32	83.14:1.32
Rand-GD	81.67:1.32	75.03:1.32	88.56:1.32

after a certain point leads to diminishing returns in performance. Further, our proposed methods perform better than rest of the methods in comparison. In general, active learning methods beat the random sampling method by a large margin and shall play a vital role in establishing the super pareto norm.

There are very few studies that have analysed the effect of diminishing returns in active learning. There are no clear answers to - when to stop an active learning procedure? or how to estimate the expected performance gain when adding certain set of data points? These could be foundation for exciting future research.

5.2 CONCLUSIONS AND FUTURE WORKS

In this study, we presented theoretical and practical perspectives of active learning. Active learning has had a rich history in machine learning and recently in deep learning. However, along the way, it lost its foundation trying to fit into varied applications with little practical benefits. We discussed the limitations of active learning specifically for image classification and segmentation. Nevertheless, we believe that the presented arguments can be extended for other applications such as natural language, healthcare, reinforcement learning and so on. In most active learning methods, oracle is incorporated only for providing annotations. In remote sensing, oracle can also play a vital role by providing domain level knowledge in specialised applications like wildlife monitoring, vulnerability analysis and so on.

In the first section, we presented the performance of ALSA on a few datasets including UC Merced and EuroSat. We used the LeNet5 and VGG16 networks for the classification tasks. We noticed that the inclusion of representativeness measures in query strategy resulted in a better performance than methods using just informativeness measures. However, the experiments can

definitely be made more comprehensive in the future by incorporating larger and complex networks, class imbalances, noisy oracles and adversarial attacks. In most of the active learning methods, oracles are assumed to be perfect and are expected to provide labels indefatigably. At the end of the day, oracles are humans and humans make mistakes. This intuition could provide a basis for future research. Further, we may also consider a binary oracle, who can only answer yes or no questions. This would definitely reduce the efforts of oracle but introduce difficulty in solving multi class classification problems.

In the second section, we presented a novel active learning method for semantic segmentation. We noticed that the active learning methods designed for image classification do not match the performance active learning methods specifically designed for semantic segmentation. The inclusion of class wise uncertainty and proportion statistics in the query strategy results in a better spatial transferability of model with very few data points. The performance of the proposed method was evaluated on the task of crop classification using satellite image time series of NDVI. We believe that similar performance shall be achieved for other applications such as point cloud segmentation, building extraction and so on. In future, experiments can be extended by including other vegetation indexes, sensor fusion or super resolution for classification.

Explicit temporal information may be incorporated into the acquisition procedure of active learning. Techniques such as dynamic time warping or fourier transform would make a potential choice. Further, temporal convolutional networks (TCN) (Lea et al., 2017) or long short term memory (LSTM) (Shi et al., 2015) may be used to improve classification and active learning results. Spatial information is also an important consideration in remote sensing applications. Measures such as spatial autocorrelation may be incorporated into the active acquisition framework. Lastly, we recommend to experiment with different patch sizes to understand the impact of small and large patch sizes on active learning.

In the final section, we presented the few shot learning setup. We discussed various meta learning techniques and their limitations for practical applications in few shot learning. We used active learning to select optimal batch of spatial tasks for MAML. We noticed a better spatial transferability of models when using MAML as opposed to vanilla gradient descent. However, the experiments were limited. In future, the proposed method could be evaluated with model based or metric based meta learning techniques. It may be combined with generative modelling or augmentation techniques. The framework could also be modified to address problems such as domain adaptation, temporal transferability or semi supervised learning. In remote sensing, meta learning may be used to achieve efficient data assimilation over different sensors, spatial resolutions or vegetation indexes.

Appendix A

Variational Inference and Monte Carlo Dropouts

A neural network is a probabilistic model. It can be denoted as $p(y|\, x, \omega)$ where x is the input data, y is the output and ω are the parameters of the neural network. If $D = \{x^{(i)}, y^{(i)}\}$, then the likelihood function can be calculated as $p(D|\, \omega) = \prod_i p(y^i|\, x^i, \omega)$. Similarly, the posterior function can be calculated as $p(\omega|\, D) \propto p(D|\, \omega)p(\omega)$ where $p(D)$ is assumed to be constant. Maximum Likehood estimate (MLE) and Maximum A Posteriori (MAP) of ω can be calculated using the Expectation Maximization (EM) algorithm.

Both MAP and MLE give a single estimate of ω. However, there is never a single way of explaining a given data distribution. If $p(\omega|\, D)$ is estimated, then we could have several estimates of a single data point by varying ω sampled from $p(\omega|\, D)$. These estimates can be integrated together to form a predictive distribution of a testing point x^* -

$$p(y^*|\, x^*, D) = \int p(y^*|\, x^*, \omega)p(\omega|\, D)d\omega \tag{A.1}$$

Again, $p(D)$ is assumed to be constant and ignored in the integration. A closed form for $p(\omega|\, D)$ does not exist for neural networks. As a result, $p(\omega|\, D)$ is approximated by a variational distribution $q(\omega|\, \theta)$. θ is estimated by minimizing the Kullback Leibler Divergence (KLD) between $q(\omega|\, \theta)$ and $p(\omega|\, D)$.

$$KL(q(\omega|\, \theta)||\, p(\omega|\, D)) = \int q(\omega|\, \theta) log \frac{q(\omega|\, \theta)}{p(\omega|\, D)} d\omega \tag{A.2}$$

$$KL(q(\omega|\, \theta)||\, p(\omega|\, D)) = E_{q(\omega|\, \theta)} log \frac{q(\omega|\, \theta)}{p(\omega|\, D)} \tag{A.3}$$

$$= E_{q(\omega|\, \theta)} log \frac{q(\omega|\, \theta)}{p(D|\, \omega)p(\omega)} p(D) \tag{A.4}$$

$$KL(q(\omega|\, \theta)||\, p(\omega|\, D)) = E_{q(\omega|\, \theta)}[log\, q(\omega|\, \theta) - log\, p(D|\, \omega) - log\, p(\omega)] + log\, p(D) \tag{A.5}$$

$$= KL(q(\omega|\, \theta)||\, p(\omega)) - E_{q(\omega|\, \theta)}[log\, p(D|\, \omega)] + log\, p(D) \tag{A.6}$$

Notice that $p(D)$ does not involve ω. So, it can be ignored when minimizing the overall KLD. Remaining term $F(D, \theta) = KL(q(\omega|\, \theta)||\, p(\omega)) - E_{q(\omega|\, \theta)}[log\, p(D|\, \omega)]$ is called the variational free energy. The negative of variational free energy is known as the evidence lower bound $L(D, \theta)$.

$$KL(q(\omega|\, \theta)||\, p(\omega|\, D)) = -L(D, \theta) + log\, p(D) \tag{A.7}$$

$$L(D, \theta) = log\, p(D) - KL(q(\omega|\, \theta)||\, p(\omega|\, D)) \tag{A.8}$$

$$L(D, \theta) \leq log\, p(D) \tag{A.9}$$

The overall KLD term can be minimized by minimizing $F(D, \theta)$ or maximizing $L(D, \theta)$. Blundell et al., (2015) use the same procedure to model weight uncertainty in neural networks. They use gaussian variational distribution to approximate the posterior. Further, they use a reparameterization trick for training the neural network with backpropogation.

Monte Carlo (MC) dropouts introduced by (Gal & Ghahramani, 2016) takes a different approach. They prove that dropouts applied to weights are mathematically equivalent to approximate variational inference. To achieve this, they make use of gaussian process approximated with monte carlo integration. Dropouts at inference time is equivalent to meaning that the variational distribution $q(\omega)$ is a set of matrices with random zero entries drawn from a bernoulli distribution. They approximate its evidence lower bound by drawing MC samples from this variational distribution and then using MC integration. This is equivalent to T forward passes of the network. Finally, they prove that the limit of evidence lower bound from MC dropout converge to the same limit as from variational inference. For the variational inference, they use a simple gaussian variational function. The predictive distribution of a testing point x^* can be approximated using MC dropouts as follows -

$$p(y^* | \, x^*, D) = \int p(y^* | \, x^*, \omega) p(\omega | \, D) d\omega \tag{A.10}$$

$$\approx \int p(y^* | \, x^*, \omega) q(\omega) d\omega \tag{A.11}$$

$$\approx \frac{1}{T} \sum_{t=1}^{T} p(y^* | \, x^*, \omega_t) \tag{A.12}$$

where $\omega_t \sim q(\omega)$.

Appendix B

Simulated Annealing

Simulated annealing (SA) is a metaheuristic search-based optimization algorithm (S. Kirkpatrick et al., 1983). It is used to find the global maxima or minima of a distribution. It is an iterative algorithm which starts with a solution S sampled at random from the given distribution. The solution then navigates around its neighborhood to escape from local maxima or minima. The algorithm gradually shifts from exploration stage in the beginning to exploitation stage towards the end. This shift is controlled by a temperature parameter T which decreases over the course of iterations (also known as the annealing schedule). At high temperatures, sub optimal solutions are likely to be selected. At low temperatures, the algorithm becomes more careful in selecting solutions and eventually converges to the global solution.

Algorithm 6 Simulated Annealing

Require: f, S, N
Set $T \leftarrow 1$
for $n = 1, ...N$ **do**:
  Sample $S^* \sim f$ in the neighborhood of S
  if $f(S) - f(S^*) > 0$ **then**:
    $S \leftarrow S^*$
  else:
    if $randu(0,1) < e^{-\frac{f(S)^* - f(S)}{T}}$ **then**:
      $S \leftarrow S^*$
  $T \leftarrow e^{-\frac{n}{N}}$
end for

There are various annealing schedules proposed in the literature. The algorithm presented in Algorithm 6 uses a negative exponential annealing schedule and finds the global minima of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The acceptance probability of a solution depends on the value of T and the difference between current and the sampled solution. If the current and the sampled solutions are S and S^* respectively, then the acceptance probability is given by -

$$p(S^* | \, S, T) = e^{-\frac{f(S)^* - f(S)}{T}} \tag{B.1}$$

Appendix C

KMeans Clustering and KMeans++ Seeding Algorithm

KMeans clustering is a widely used unsupervised clustering algorithm. The overall objective of the algorithm is to separate data points into k clusters such that the intra cluster variation is minimum. Each cluster is represented by a centroid which is calculated as the mean of all data points belonging to that cluster. Data points are assigned to clusters using the nearest centroid algorithm. Intra cluster variation is the distance of data points from their cluster centroids. So, the objective function for KMeans clustering is defined by -

$$c_1, c_2, ...c_k = \underset{c}{\text{argmin}} \sum_{j=1}^{k} \sum_{x \in c_j} ||x - c_j||^2 \tag{C.1}$$

The above objective function is NP-hard as we need to iterate through all possible ${}^nC_k$ centroid combinations. Thus, we use the Lloyd's algorithm[1] as an approximation to the KMeans.

Algorithm 7 KMeans Clustering

Require: $k, \{x_1, x_2, ...x_n\}$
Random Sample: $\{c_1, c_2, ...c_k\} \in \{x_1, x_2, ...x_n\}$
while not done:
  for $i = 1, ...n$ **do:**
    for $j = 1, ...k$ **do:**

$$z_{ij} \leftarrow \begin{cases} 1, & \text{if } j = \underset{p}{\text{argmin}} \, ||x_i - c_p||^2 \\ 0, & \text{otherwise} \end{cases}$$

    end for
  end for
  for $j = 1, ...k$ **do:**
    $n_j \leftarrow \sum_i z_{ij}$
    $c_j \leftarrow \frac{1}{n_j} \sum_i z_{ij} x_i$
  end for
end while

It uses the nearest centroid algorithm to assign data points to k (user defined) cluster centroids. These centroids are randomly selected (in the beginning of the algorithm) from the available data points. Through an iterative procedure, intra cluster variation is reduced by updating the cluster centroids and re running the nearest centroid algorithm. Algorithm 7 presents the KMeans

[1] From here, by referring to KMeans, we refer to the Lloyd's algorithm (common in most of the existing literature).

algorithm on data points $\{x_1, x_2, ...x_n\} \in \mathbb{R}^n$. As it is evident that the KMeans algorithm does not guarantee to output the global minima of intra cluster variations. The final clusters depend on the selection of the initial cluster centroids.

Recently, Arthur et al., (2007) proposed KMeans++ algorithm for seeding the initial cluster centroids in KMeans. It begins with a randomely selected centroid and then iteratively selects the next centroid which has the largest distance from existing centroids. The pseudo code for KMeans++ seeding algorithm is presented in Algorithm 8. Lastly, it is worth noting that the final cluster centroids generated by the KMeans algorithm does not belong to the original dataset. The K-Medoids clustering algorithm can be used to obtain cluster centroids that belong to the original dataset. However, the K-Medoids algorithm is much more computationally expensive than K-Means and therefore is not suitable for practical applications.

Algorithm 8 KMeans++ Seeding Algorithm

Require: $k, \{x_1, x_2, ...x_n\}$
Random Sample: $c_1 \in \{x_1, x_2, ...x_n\}$
Set: $m \leftarrow 1$
while $m \leq k$ **do**:
  for $i = 1, ...n$ **do**:
    $d_i = \min_{j=1:m} ||x_i - c_j||^2$
  end for
  index $\sim p_j(\frac{d_j^2}{\sum_i d_i^2})$
  $m \leftarrow m + 1$
  $c_m \leftarrow x_{index}$
end while

Appendix D

Datasets

D.1 MNIST

Access Link: https://www.tensorflow.org/datasets/catalog/mnist.
Train-Val-Test: 75:10:15

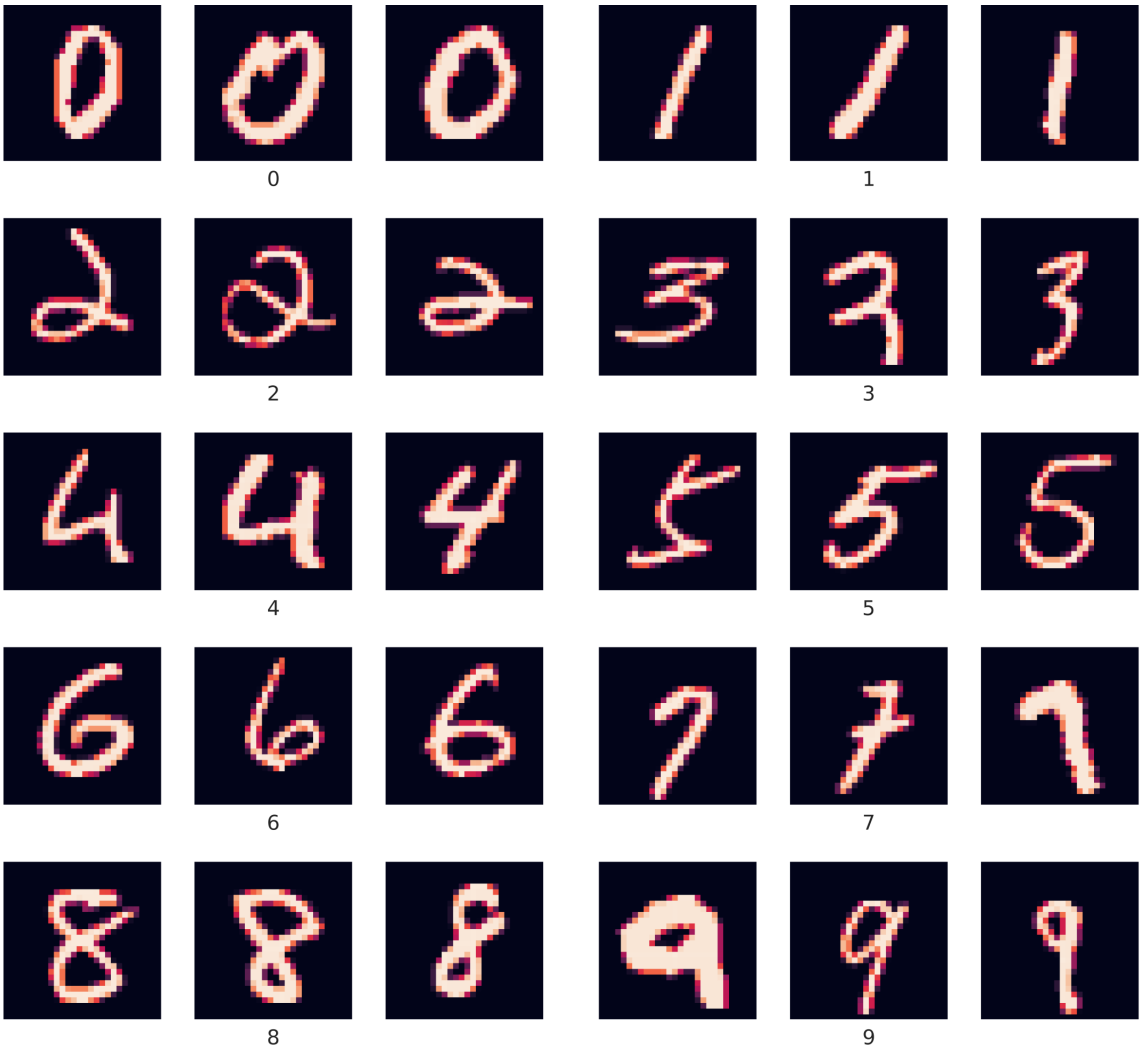


Figure D.1: The Mnist dataset has 10 classes with 7000 grayscale images per class.

D.2 FASHION MNIST

Access Link: https://www.tensorflow.org/datasets/catalog/fashion_mnist.
Train-Val-Test: 75:10:15

Figure D.2: The Fashion Mnist dataset has 10 classes with 7000 grayscale images per class.

D.3 CIFAR 10

Access Link: https://www.tensorflow.org/datasets/catalog/cifar10.
Train-Val-Test: 75:10:15

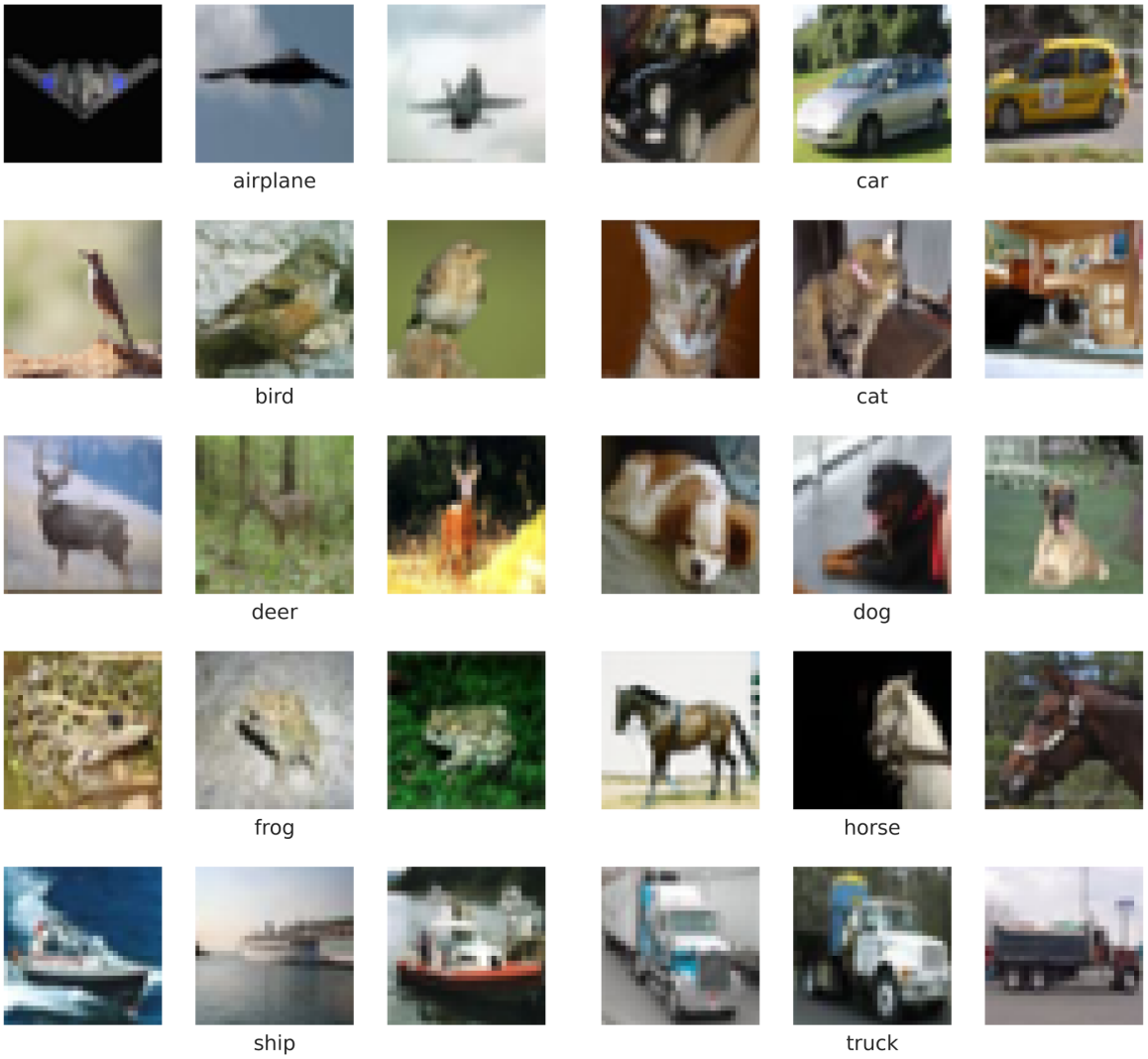
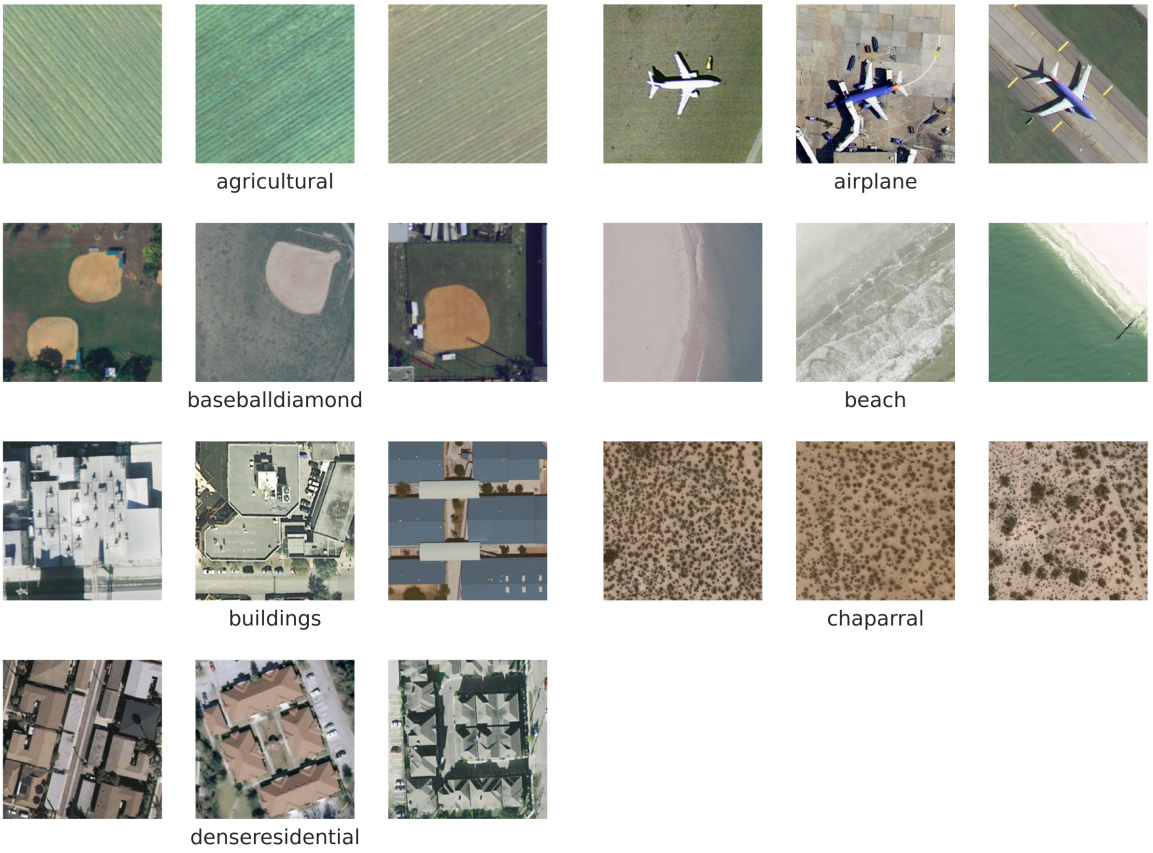


Figure D.3: The Cifar10 dataset has 10 classes with 6000 RGB images per class.

D.4 UC MERCED

Access Link: http://weegee.vision.ucmerced.edu/datasets/landuse.html.
Train-Val-Test: 75:10:15



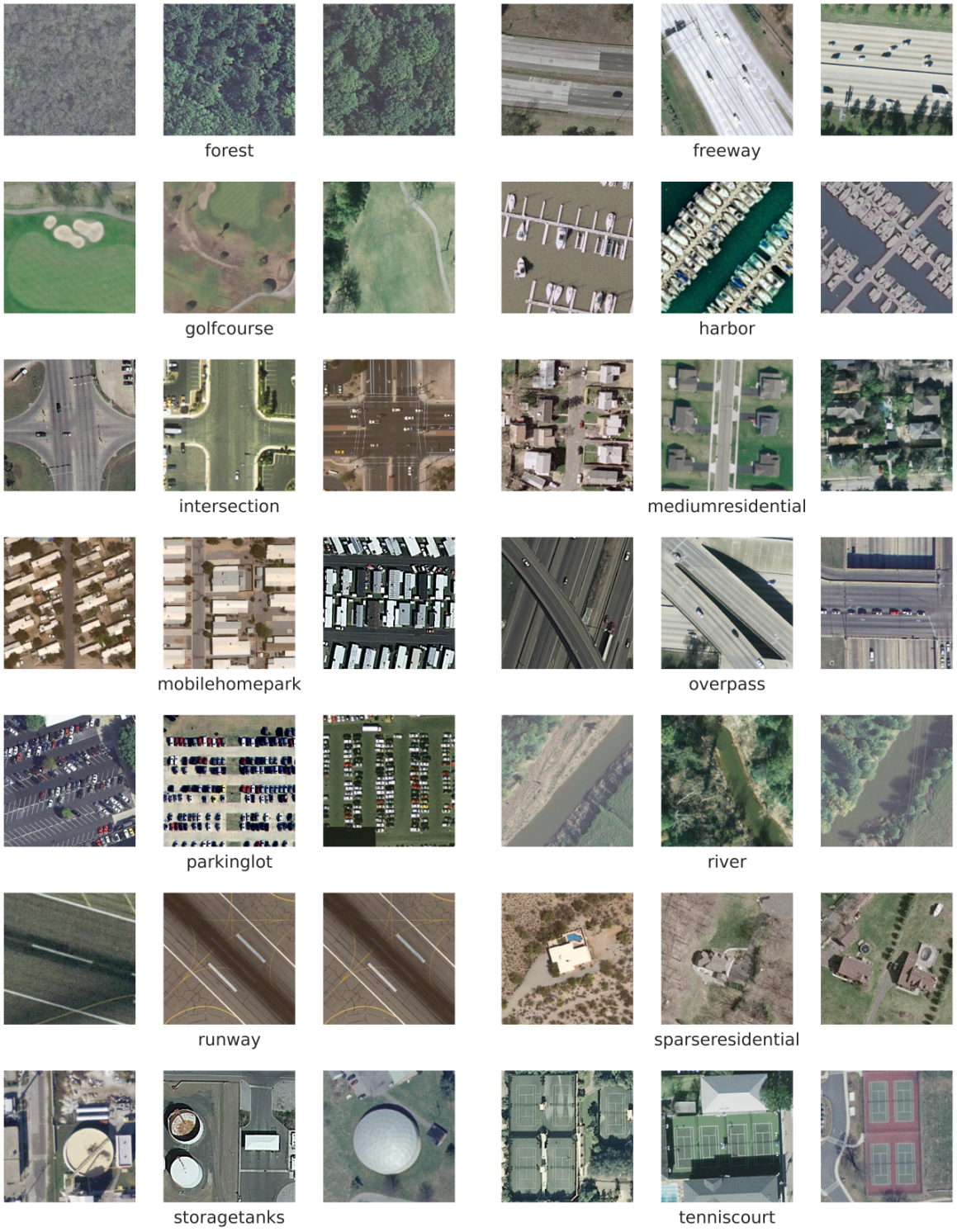


Figure D.4: The UC Merced dataset has 21 classes with 100 RGB images per class.

D.5 EUROSAT

Access Link: https://github.com/phelber/EuroSAT.
Train-Val-Test: 75:10:15

Figure D.5: The EuroSat dataset has 10 classes with varying number of RGB images per class.

Appendix E

Comparison of Loss Functions for Semantic Segmentation

Loss functions are used as an objective to evaluate how close or far a predicted value is from its true value. The goal of an optimization algorithm is to find an optimal set of weights that shall lead to the least prediction loss. Majority of the loss functions cannot be optimized efficiently in practice. As a result, surrogate loss functions are defined that make learning mathematically convenient and efficient. In literature, the term loss function implicitly refers to surrogate loss function. This is done to maintain good readability (which we do so henceforth). For the rest of this section, we denote the input data as x, $\hat{y}$ as the output, y as the true data and ω as the parameters of the neural network.

The most common loss function for classification is the cross entropy loss (Good, 1952). It optimizes the accuracy of classification. However, it is poor at handling class imbalances in the training dataset. It is given by -

$$l_{CE}(\hat{y}, y) = \sum_{c} -y_c log(p_\omega(\hat{y} = c)) \tag{E.1}$$

The weighted cross entropy loss introduces weights for each class in the cross entropy loss function. It is good at handling class imbalances but fails to capture difference between high confidence and low confidence predictions. Let β denote the array of weights, then -

$$l_{WCE}(\hat{y}, y) = \sum_{c} -\beta_c y_c log(p_\omega(\hat{y} = c)) \tag{E.2}$$

The focal loss (T.-Y. Lin et al., 2017) penalizes high confidence incorrect predictions more than low confidence incorrect predictions. It does so by introducing an exponential parameter γ -

$$l_{FL}(\hat{y}, y) = \sum_{c} -\beta_c y_c (1 - p_\omega(\hat{y} = c))^\gamma log(p_\omega(\hat{y} = c)) \tag{E.3}$$

The above discussed loss functions try to optimize the accuracy of classification. However, accuracy is not an appropriate metric for evaluating the performance of a semantic segmentation task. It ignores the minority and foreground classes which are commonly encountered in segmentation tasks. The F1 and mIoU score correctly measure the overlap between the predicted and true maps.

Jaccard distance directly optimizes the mIoU score and the dice loss optimizes the F1 score. Jaccard distance is given by -

$$l_{JC}(\hat{y}, y) = 1 - \frac{y\hat{y} + \epsilon}{y + \hat{y} - y\hat{y} + \epsilon} \tag{E.4}$$

where ϵ is a smoothing constant to avoid mathematical overflows. Dice loss can be calculate from the jaccard distance as -

$$l_{DC}(\hat{y}, y) = 2\frac{l_{JC}(\hat{y}, y)}{1 + l_{JC}(\hat{y}, y)} \tag{E.5}$$

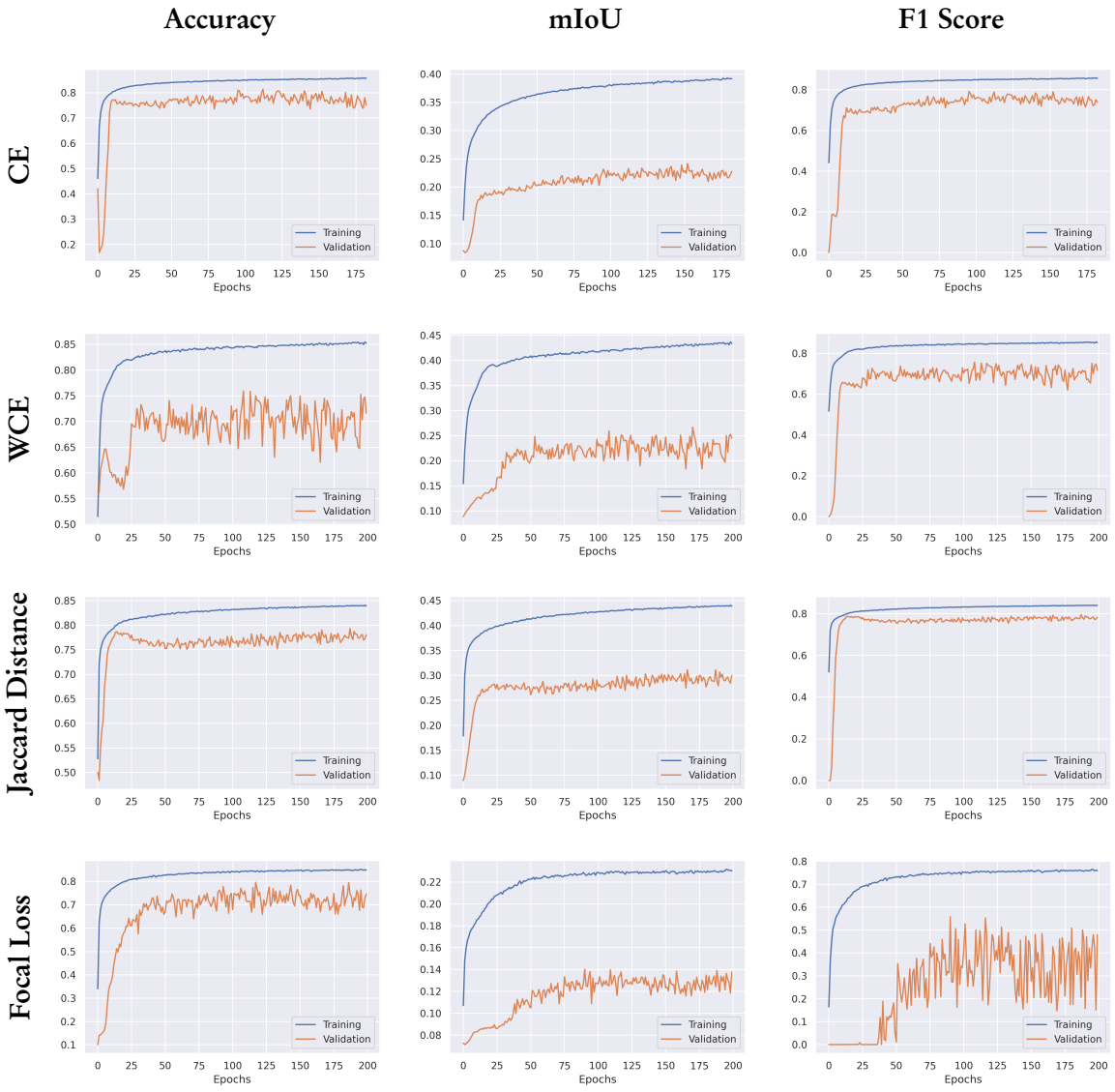


Figure E.1: The training and validation curves of accuracy, mIoU and F1 score when the segmenting network is trained using four different loss functions.

Appendix F

Convolutional Variational Autoencoder

An autoencoder is a neural network consisting of stacked encoder-decoder neural networks. The encoder learns to embed given data to a low dimensional space, while the decoder learns to reconstruct the data from this embedding. This kind of network is used for unsupervised dimensionality reduction, feature extraction or embedding learning. Variational autoencoder (VAE) (Kingma & Welling, 2014) is a type of autoencoder that can generate new data after learning to embed given data to a latent space. We denote x as input data, z as the variable for latent space and $x^{'}$ as reconstructed data. New data can be generated directly if $p(z|x)$ is known. Since it is not possible to directly estimate $p(z|x)$, a VAE encoder learns the variational distribution $q(z|x)$, that serves as an approximation to $p(z|x)$. The variational distribution $q(z|x)$ is assumed to be a gaussian distribution with mean $\mu(x)$ and standard deviation $\sigma(x)$. The training procedure of VAE tries to learn the optimal value of these parameters. New data can be generated by sampling from the distribution $\mathcal{N}(\mu(x), \sigma(x))$ and passing it through the decoder. VAE's decoder learns the distribution $q(x^{'}|z)$ during the training.

The learning procedure of VAE consists of two losses. The first loss minimizes the reconstruction error between x and $x^{'}$. This is generally the L2 loss. The other loss minimizes the KLD between $q(z|x)$ and $p(z|x)$. This is similar to the variational inference procedure presented in A. The prior probability $p(z)$ is set to the gaussian normal distribution $\mathcal{N}(0, 1)$. This is done to ensure embedding all x's to the same euclidean space. The mean $\mu(x)$ and the standard deviation $\sigma(x)$ is provided by the last layer of the encoder. This last layer contains twice the number of neurons than the required size of latent dimension. The first half computes $\mu(x)$ and the second half computes $\sigma(x)$. During the forward propagation, z is sampled from the distribution $\mathcal{N}(\mu(x), \sigma(x))$. During backpropagation, the reparameterization trick is used which represents z as $\mu(x) + \epsilon.\sigma(x)$, where $\epsilon \sim \mathcal{N}(0, 1)$.

A convolutional VAE consists of fully convolutional layers within its encoder and decoder. It learns to generate images from given reference set of images. In this study, we use a set of 4 convolutional layers inside the encoder and 4 transposed convolutional layers inside the decoder of VAE. The latent dimension is set to 500. We use the Adam optimizer with a learning rate of 1e-4. The VAE is trained for 50 epochs.

List of References

Abraham, A., & Dreyfus-Schmidt, L. (2021). Sample Noise Impact on Active Learning. http://arxiv.org/abs/2109.01372 (cit. on p. 17)

Alajaji, D., Alhichri, H. S., Ammour, N., & Alajlan, N. (2020). Few-shot learning for remote sensing scene classification. *2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS)*, 81–84. https://doi.org/10.1109/M2GARSS47143.2020.9105154 (cit. on p. 35)

Antoniou, A., Storkey, A., & Edwards, H. (2019). How to train your MAML. *7th International Conference on Learning Representations, ICLR 2019* (cit. on p. 37).

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, *07-09-Janu*, 1027–1035. https://doi.org/10.5555/1283383 (cit. on p. 54)

Ash, J. T., Zhang, C., Krishnamurthy, A., Langford, J., & Agarwal, A. (2019). Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. http://arxiv.org/abs/1906.03671 (cit. on pp. 8, 10, 15)

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. https://doi.org/10.48550/ARXIV.1511.00561. (Cit. on p. 29)

Balaji, Y., Sankaranarayanan, S., & Chellappa, R. (2018). Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, *2018-Decem*, 998–1008 (cit. on p. 38).

Ball, J. E., Anderson, D. T., & Chan, C. S. (2017). A comprehensive survey of deep learning in remote sensing: Theories, tools and challenges for the community. *arXiv*, *11*(May). https://doi.org/10.1117/1.jrs.11.042609 (cit. on p. 1)

Bischke, B., Helber, P., Folz, J., Borth, D., & Dengel, A. (2019). Multi-Task Learning for Segmentation of Building Footprints with Deep Neural Networks. *Proceedings - International Conference on Image Processing, ICIP*, *2019-Septe*, 1480–1484. https://doi.org/10.1109/ICIP.2019.8803050 (cit. on p. 2)

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. *32nd International Conference on Machine Learning, ICML 2015*, *2*, 1613–1622. https://arxiv.org/abs/1505.05424v2 (cit. on p. 50)

Brandt, J. (2019). Spatio-temporal crop classification of low-resolution satellite imagery with capsule layers and distributed attention. http://arxiv.org/abs/1904.10130 (cit. on p. 26)

Bruzzone, L., & Persello, C. (2009). Active learning for classification of remote sensing images. *International Geoscience and Remote Sensing Symposium (IGARSS)*, *3*, 693–696. https://doi.org/10.1109/IGARSS.2009.5417857 (cit. on p. 1)

Chandra, A. L., Desai, S. V., Devaguptapu, C., & Balasubramanian, V. N. (2020). On Initial Pools for Deep Active Learning. *Proceedings of Machine Learning Research*, *148*, 14–32. http://arxiv.org/abs/2011.14696 (cit. on p. 7)

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. https://doi.org/10.48550/ARXIV.1706.05587. (Cit. on p. 25)

Chen, X., Yuan, Y., Zeng, G., & Wang, J. (2021). Semi-Supervised Semantic Segmentation with Cross Pseudo Supervision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2613–2622. https://doi.org/10.1109/CVPR46437.2021.00264 (cit. on p. 25)

Cho, J. H., Mall, U., Bala, K., & Hariharan, B. (2021). Picie: Unsupervised Semantic Segmentation using Invariance and Equivariance in Clustering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 16789–16799. https://doi.org/10.1109/CVPR46437.2021.01652 (cit. on p. 25)

Citovsky, G., DeSalvo, G., Gentile, C., Karydas, L., Rajagopalan, A., Rostamizadeh, A., & Kumar, S. (2021). Batch Active Learning at Scale. http://arxiv.org/abs/2107.14263 (cit. on pp. 8, 10, 15)

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145. https://doi.org/10.1613/jair.295 (cit. on p. 7)

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 3213–3223. https://doi.org/10.1109/CVPR.2016.350 (cit. on p. 35)

Davari, A., Ozkan, H. C., Maier, A., & Riess, C. (2019). Fast and efficient limited data hyperspectral remote sensing image classification via gmm-based synthetic samples. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *12*(7), 2107–2120. https://doi.org/10.1109/JSTARS.2019.2916495 (cit. on p. 2)

Débonnaire, N., Stumpf, A., & Puissant, A. (2016). Spatio-Temporal Clustering and Active Learning for Change Classification in Satellite Image Time Series. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *9*(8), 3642–3650. https://doi.org/10.1109/JSTARS.2016.2525940 (cit. on p. 2)

Demir, B., Persello, C., & Bruzzone, L. (2011). Batch-mode active-learning methods for the interactive classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *49*(3), 1014–1031. https://doi.org/10.1109/TGRS.2010.2072929 (cit. on p. 2)

Demir, B., Minello, L., & Bruzzone, L. (2014). Definition of effective training sets for supervised classification of remote sensing images by a novel cost-sensitive active learning method. *IEEE Transactions on Geoscience and Remote Sensing*, *52*(2), 1272–1284. https://doi.org/10.1109/TGRS.2013.2249522 (cit. on p. 25)

Dhariwal, P., & Nichol, A. (2021). Diffusion Models Beat GANs on Image Synthesis. http://arxiv.org/abs/2105.05233 (cit. on p. 2)

Ducoffe, M., & Precioso, F. (2018). Adversarial Active Learning for Deep Networks: a Margin Based Approach. http://arxiv.org/abs/1802.09841 (cit. on p. 11)

Farquhar, S., Gal, Y., & Rainforth, T. (2021). On statistical bias in active learning: How and when to fix it. https://doi.org/10.48550/ARXIV.2101.11665. (Cit. on p. 13)

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision 2004 59:2*, *59*(2), 167–181. https://doi.org/10.1023/B:VISI.0000022288.19776.77 (cit. on p. 30)

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *34th International Conference on Machine Learning, ICML 2017*, *3*, 1856–1868. http://arxiv.org/abs/1703.03400 (cit. on pp. 3, 37)

Gal, Y., & Ghahramani, Z. (2015). Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference. http://arxiv.org/abs/1506.02158 (cit. on p. 7)

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *33rd International Conference on Machine Learning, ICML 2016*, *3*, 1651–1660. http://yarin.co. (cit. on pp. 7, 50)

Gal, Y., Islam, R., & Ghahramani, Z. (2017). Deep Bayesian active learning with image data. *34th International Conference on Machine Learning, ICML 2017*, *3*, 1923–1932. https://doi.org/10.17863/CAM.11070 (cit. on pp. 2, 9, 15)

Geiß, C., Thoma, M., & Taubenböck, H. (2018). Cost-Sensitive Multitask Active Learning for Characterization of Urban Environments with Remote Sensing. *IEEE Geoscience and Remote Sensing Letters*, *15*(6), 922–926. https://doi.org/10.1109/LGRS.2018.2813436 (cit. on p. 25)

Ghiasi, G., Lin, T.-Y., & Le, Q. V. (2018). Dropblock: A regularization method for convolutional networks. https://doi.org/10.48550/ARXIV.1810.12890. (Cit. on p. 30)

Good, I. J. (1952). Rational Decisions on JSTOR. *Journal of the Royal Statistical Society Series B (…* https://www.jstor.org/stable/2984087?seq=1%20http://www.jstor.org/stable/2984087%5C%5Cnpapers3://publication/doi/10.2307/2984087 (cit. on p. 61)

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622 (cit. on p. 2)

Harshvardhan, G., Gourisaria, M. K., Pandey, M., & Rautaray, S. S. (2020). A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, *38*, 100285. https://doi.org/10.1016/j.cosrev.2020.100285 (cit. on p. 2)

He, Z., Liu, H., Wang, Y., & Hu, J. (2017). Generative adversarial networks-based semi-supervised learning for hyperspectral image classification. *Remote Sensing*, *9*(10), 1042. https://doi.org/10.3390/rs9101042 (cit. on p. 2)

Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *12*(7), 2217–2226. https://doi.org/10.1109/JSTARS.2019.2918242 (cit. on p. 21)

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. https://doi.org/10.48550/ARXIV.1503.02531. (Cit. on p. 1)

Hospedales, T. M., Antoniou, A., Micaelli, P., & Storkey, A. J. (2021). Meta-Learning in Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2021.3079209 (cit. on pp. 2, 37)

Houlsby, N., Huszár, F., Ghahramani, Z., & Lengyel, M. (2011). Bayesian Active Learning for Classification and Preference Learning. http://arxiv.org/abs/1112.5745 (cit. on pp. 9, 15, 31)

Hsu, W. N., & Lin, H. T. (2015). Active learning by learning. *Proceedings of the National Conference on Artificial Intelligence*, *4*, 2659–2665. www.aaai.org (cit. on p. 11)

Huang, S. J., Jin, R., & Zhou, Z. H. (2014). Active Learning by Querying Informative and Representative Examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(10), 1936–1949. https://doi.org/10.1109/TPAMI.2014.2307881 (cit. on p. 10)

Ibrahim, E. S., Rufin, P., Nill, L., Kamali, B., Nendel, C., & Hostert, P. (2021). Mapping crop types and cropping systems in nigeria with sentinel-2 imagery. *Remote Sensing*, *13*(17), 1–24. https://doi.org/10.3390/rs13173523 (cit. on p. 26)

Jamal, M. A., Qi, G.-J., & Shah, M. (2018). Task-agnostic meta-learning for few-shot learning. https://doi.org/10.48550/ARXIV.1805.07722. (Cit. on p. 38)

James, & others, M. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, *1*(14), 281–297 (cit. on p. 12).

Ji, X., Vedaldi, A., & Henriques, J. (2019). Invariant information clustering for unsupervised image classification and segmentation. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 9864–9873. https://doi.org/10.1109/ICCV.2019.00996 (cit. on p. 25)

Kaddour, J., Sæmundsson, S., & Deisenroth, M. P. (2020). Probabilistic active meta-learning. https://doi.org/10.48550/ARXIV.2007.08949. (Cit. on p. 38)

Kapoor, A., Grauman, K., Urtasun, R., & Darrell, T. (2007). Active learning with Gaussian processes for object categorization. *Proceedings of the IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV.2007.4408844 (cit. on p. 9)

Kim, K., Park, D., Kim, K. I., & Chun, S. Y. (2021). Task-Aware Variational Adversarial Active Learning, 8162–8171. https://doi.org/10.1109/cvpr46437.2021.00807 (cit. on p. 10)

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*. https://arxiv.org/abs/1312.6114v10%20http://arxiv.org/abs/1312.6114 (cit. on pp. 2, 30, 63)

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, *114*(13), 3521–3526. https://doi.org/10.1073/pnas.1611835114 (cit. on p. 43)

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671 (cit. on p. 51)

Kirsch, A., van Amersfoort, J., & Gal, Y. (2019). BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning. *Advances in Neural Information Processing Systems*, *32*. http://arxiv.org/abs/1906.08158 (cit. on pp. 8, 10)

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. *Cs.Toronto.Edu*, 1–58 (cit. on p. 14).

Kushnir, D., & Venturi, L. (2020). Diffusion-based Deep Active Learning. http://arxiv.org/abs/2003.10339 (cit. on p. 11)

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 1003–1012. https://doi.org/10.1109/CVPR.2017.113 (cit. on p. 47)

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2323. https://doi.org/10.1109/5.726791 (cit. on p. 14)

LeCun, Y., Cortes, C., & Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, *2* (cit. on p. 14).

Lee, H. B., Nam, T., Yang, E., & Hwang, S. J. (2019). Meta dropout: Learning to perturb features for generalization. https://doi.org/10.48550/ARXIV.1905.12914. (Cit. on p. 38)

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. https://doi.org/10.48550/ARXIV.1708.02002. (Cit. on p. 61)

Lin, Y., Vosselman, G., Cao, Y., & Yang, M. Y. (2020). Active and incremental learning for semantic ALS point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, *169*, 73–92. https://doi.org/10.1016/j.isprsjprs.2020.09.003 (cit. on pp. 25, 30)

Liu, S., & Shi, Q. (2020). Multitask Deep Learning with Spectral Knowledge for Hyperspectral Image Classification. *IEEE Geoscience and Remote Sensing Letters*, *17*(12), 2110–2114. https://doi.org/10.1109/LGRS.2019.2962768 (cit. on pp. 2, 11)

Liu, X., Zhou, Y., Zhao, J., Yao, R., Liu, B., & Zheng, Y. (2019). Siamese Convolutional Neural Networks for Remote Sensing Scene Classification. *IEEE Geoscience and Remote Sensing Letters*, *16*(8), 1200–1204. https://doi.org/10.1109/LGRS.2019.2894399 (cit. on p. 3)

Mackowiak, R., Lenz, P., Ghori, O., Diego, F., Lange, O., & Rother, C. (2019). CEREALS - Cost-Effective REgion-based Active Learning for Semantic Segmentation. *British Machine Vision Conference 2018, BMVC 2018*. https://arxiv.org/abs/1810.09726v1 (cit. on p. 25)

Margatina, K., Vernikos, G., Barrault, L., & Aletras, N. (2021). Active learning by acquiring contrastive examples. https://doi.org/10.48550/ARXIV.2109.03764. (Cit. on p. 11)

Masarczyk, W., & Tautkute, I. (2020). Reducing catastrophic forgetting with learning on synthetic data. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1019–1024. https://doi.org/10.1109/CVPRW50498.2020.00134 (cit. on p. 43)

Mehta, B., Diaz, M., Golemo, F., Pal, C. J., & Paull, L. (2019). Active domain randomization. https://doi.org/10.48550/ARXIV.1904.04762. (Cit. on p. 38)

Munjal, P., Hayat, N., Hayat, M., Sourati, J., & Khan, S. (2020). Towards Robust and Reproducible Active Learning Using Neural Networks. https://github.com/hysts/pytorch_shake_shake%20http://arxiv.org/abs/2002.09564 (cit. on pp. 3, 8)

Munkhdalai, T., & Yu, H. (2017). Meta Networks. *34th International Conference on Machine Learning, ICML 2017*, *5*, 3933–3943. https://doi.org/10.48550/arxiv.1703.00837 (cit. on p. 37)

Niazmardi, S., Homayouni, S., & Safari, A. (2019). A computationally efficient multi-domain active learning method for crop mapping using satellite image time-series. *International Journal of Remote Sensing*, *40*(16), 6383–6394. https://doi.org/10.1080/01431161.2019.1591648 (cit. on p. 26)

Nichol, A., Achiam, J., & Schulman, J. (2018). On First-Order Meta-Learning Algorithms. http://arxiv.org/abs/1803.02999 (cit. on p. 3)

Nowakowski, A., Mrziglod, J., Spiller, D., Bonifacio, R., Ferrari, I., Mathieu, P. P., Garcia-Herranz, M., & Kim, D. H. (2021). Crop type mapping by using transfer learning. *International Journal of Applied Earth Observation and Geoinformation*, *98*, 102313. https://doi.org/10.1016/j.jag.2021.102313 (cit. on p. 26)

Ouali, Y., Hudelot, C., & Tami, M. (2020). Semi-supervised semantic segmentation with cross-consistency training. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 12671–12681. https://doi.org/10.1109/CVPR42600.2020.01269 (cit. on p. 25)

Pasolli, E., Melgani, F., Tuia, D., Pacifici, F., & Emery, W. J. (2011). Improving active learning methods using spatial information. *International Geoscience and Remote Sensing Symposium (IGARSS)*, (May 2014), 3923–3926. https://doi.org/10.1109/IGARSS.2011.6050089 (cit. on p. 2)

Persello, C., Boularias, A., Dalponte, M., Gobakken, T., Naesset, E., & Scholkopf, B. (2014). Cost-sensitive active learning with lookahead: Optimizing field surveys for remote sensing

data classification. *IEEE transactions on geoscience and remote sensing*, *52*(10), 6652–6664. https://doi.org/10.1109/TGRS.2014.2300189 (cit. on p. 25)

Pop, R., & Fulop, P. (2018). Deep Ensemble Bayesian Active Learning : Addressing the Mode Collapse issue in Monte Carlo dropout via Ensembles. *arXiv*. http://arxiv.org/abs/1811.03897 (cit. on pp. 8, 10)

Qu, Z., Du, J., Cao, Y., Guan, Q., & Zhao, P. (2020). Deep Active Learning for Remote Sensing Object Detection. http://arxiv.org/abs/2003.08793 (cit. on p. 2)

Ramdas, A., Poczos, B., Singh, A., & Wasserman, L. (2014). An analysis of active learning with uniform feature noise. *Journal of Machine Learning Research*, *33*, 805–813 (cit. on p. 19).

Ravi, S., & Larochelle, H. (2017). Optimization as a model for few-shot learning. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (cit. on p. 37).

Rodríguez, A. C., D'Aronco, S., Schindler, K., & Wegner, J. D. (2021). Mapping oil palm density at country scale: An active learning approach. *Remote Sensing of Environment*, *261*, 112479. https://doi.org/10.1016/j.rse.2021.112479 (cit. on p. 2)

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. https://doi.org/10.48550/ARXIV.1505.04597. (Cit. on p. 25)

Ruswurm, M., Wang, S., Korner, M., & Lobell, D. (2020). Meta-learning for few-shot land cover classification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, *2020-June*, 788–796. https://doi.org/10.1109/CVPRW50498.2020.00108 (cit. on pp. 3, 37, 44)

Růžička, V., D'Aronco, S., Wegner, J. D., & Schindler, K. (2020). Deep Active Learning in Remote Sensing for data efficient Change Detection. *CEUR Workshop Proceedings*, *2766*. http://arxiv.org/abs/2008.11201 (cit. on p. 2)

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., & Lillicrap, T. (2016). Meta-Learning with Memory-Augmented Neural Networks Google DeepMind (cit. on p. 37).

Scheffer, T., Decomain, C., & Wrobel, S. (2001). Active hidden markov models for information extraction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2189*, 309–318. https://doi.org/10.1007/3-540-44816-0_31 (cit. on p. 8)

Sener, O., & Savarese, S. (2018). Active learning for convolutional neural networks: A core-set approach. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. http://arxiv.org/abs/1708.00489 (cit. on pp. 8, 10, 15)

Settles, B. (2010). Active Learning Literature Survey. *Machine Learning*, *15*(2), 201–221. https://doi.org/10.1.1.167.4245 (cit. on pp. 2, 9)

Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, *2015-Janua*, 802–810 (cit. on p. 47).

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1), 1–48. https://doi.org/10.1186/s40537-019-0197-0 (cit. on p. 2)

Siddiqui, Y., Valentin, J., & Nießner, M. (2020). Viewal: Active learning with viewpoint entropy for semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 9430–9440. https://doi.org/10.1109/CVPR42600.2020.00945 (cit. on p. 25)

Sinha, S., Ebrahimi, S., & Darrell, T. (2019). Variational adversarial active learning. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 5971–5980. https://doi.org/10.1109/ICCV.2019.00607 (cit. on p. 11)

Snell, J., Swersky, K., & Zemel, R. S. (2017). Prototypical Networks for Few-shot Learning. *Advances in Neural Information Processing Systems*, *2017-December*, 4078–4088. http://arxiv.org/abs/1703.05175 (cit. on pp. 3, 37)

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., & Hospedales, T. M. (2018). Learning to Compare: Relation Network for Few-Shot Learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1199–1208. https://doi.org/10.1109/CVPR.2018.00131 (cit. on pp. 3, 37)

Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. https://doi.org/10.48550/ARXIV.1905.11946 (cit. on p. 25)

Tang, H., Li, Y., Han, X., Huang, Q., & Xie, W. (2020). A Spatial-Spectral Prototypical Network for Hyperspectral Remote Sensing Image. *IEEE Geoscience and Remote Sensing Letters*, *17*(1), 167–171. https://doi.org/10.1109/LGRS.2019.2916083 (cit. on p. 3)

Tong, S., & Koller, D. (2002). Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, *1*, 45–66. https://doi.org/10.1162/153244302760185243 (cit. on p. 9)

Torralba, A., Russell, B. C., & Yuen, J. (2010). Labelme: Online image annotation and applications. *Proceedings of the IEEE*, *98*, 1467–1484 (cit. on p. 23).

Tran, T., Do, T. T., Reid, I., & Carneiro, G. (2019). Bayesian generative active deep learning. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*, 10969–10978 (cit. on p. 11).

Tseng, G., Kerner, H., Nakalembe, C., & Becker-Reshef, I. (2021). Learning to predict crop type from heterogeneous sparse labels using meta-learning. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1111–1120. https://doi.org/10.1109/CVPRW53098.2021.00122 (cit. on pp. 26, 37, 43)

Tseng, G., Kerner, H., & Rolnick, D. (2022). Timl: Task-informed meta-learning for agriculture. https://doi.org/10.48550/ARXIV.2202.02124. (Cit. on p. 37)

van der Spoel, E., Rozing, M. P., Houwing-Duistermaat, J. J., Eline Slagboom, P., Beekman, M., de Craen, A. J. M., Westendorp, R. G. J., & van Heemst, D. (2015). Siamese Neural Networks for One-Shot Image Recognition. *ICML - Deep Learning Workshop*, *7*(11), 956–963 (cit. on pp. 3, 37).

Vineeth, R., & Jain, S. (2021). Efficacy of bayesian neural networks in active learning. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2601–2609. https://doi.org/10.1109/CVPRW53098.2021.00294 (cit. on p. 7)

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 3637–3645. https://arxiv.org/abs/1606.04080v2 (cit. on pp. 3, 37)

Vosselman, G., Coenen, M., & Rottensteiner, F. (2017). Contextual segment-based classification of airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *128*, 354–371. https://doi.org/10.1016/J.ISPRSJPRS.2017.03.010 (cit. on p. 25)

Vuolo, F., Neuwirth, M., Immitzer, M., Atzberger, C., & Ng, W. T. (2018). How much does multi-temporal Sentinel-2 data improve crop type classification? *International Journal of Applied Earth Observation and Geoinformation*, *72*, 122–130. https://doi.org/10.1016/j.jag.2018.06.007 (cit. on p. 26)

Waldner, F., Abelleyra, D. D., Verón, S. R., Zhang, M., Wu, B., Plotnikov, D., Bartalev, S., Lavreniuk, M., Skakun, S., Kussul, N., Maire, G. L., Dupuy, S., Jarvis, I., & Defourny, P. (2016). Towards a set of agrosystem-specific cropland mapping methods to address the global cropland diversity. *http://dx.doi.org/10.1080/01431161.2016.1194545*, *37*(14), 3196–3231. https://doi.org/10.1080/01431161.2016.1194545 (cit. on p. 26)

Wang, K., Zhang, D., Li, Y., Zhang, R., & Lin, L. (2017). Cost-Effective Active Learning for Deep Image Classification. *IEEE Transactions on Circuits and Systems for Video Technology*, *27*(12), 2591–2600. https://doi.org/10.1109/TCSVT.2016.2589879 (cit. on p. 9)

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Computing Surveys*, *53*(3), 1–34. https://doi.org/10.1145/3386252 (cit. on pp. 2, 35)

Wei, K., Iyer, R., & Bilmes, J. (2015). Submodularity in data subset selection and active learning. *32nd International Conference on Machine Learning, ICML 2015*, *3*, 1954–1963 (cit. on p. 10).

Weikmann, G., Paris, C., & Bruzzone, L. (2021). TimeSen2Crop: A Million Labeled Samples Dataset of Sentinel 2 Image Time Series for Crop-Type Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 4699–4708. https://doi.org/10.1109/JSTARS.2021.3073965 (cit. on p. 26)

Woo, J. O. (2021). BABA: Beta Approximation for Bayesian Active Learning. http://arxiv.org/abs/2105.14559 (cit. on pp. 8, 9)

Wu, T.-H., Liu, Y.-C., Huang, Y.-K., Lee, H.-Y., Su, H.-T., Huang, P.-C., & Hsu, W. H. (2022). ReDAL: Region-based and Diversity-aware Active Learning for Point Cloud Semantic Segmentation, 15490–15499. https://doi.org/10.1109/iccv48922.2021.01522 (cit. on p. 25)

Xia, G. S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., Zhang, L., & Lu, X. (2017). AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(7), 3965–3981. https://doi.org/10.1109/TGRS.2017.2685945 (cit. on p. 25)

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *CoRR*, *abs/1708.07747*. http://arxiv.org/abs/1708.07747 (cit. on p. 14)

Xie, M., Jean, N., Burke, M., Lobell, D., & Ermon, S. (2016). Transfer learning from deep features for remote sensing and poverty mapping. *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 3929–3935. www.aaai.org (cit. on p. 2)

Xie, S., Feng, Z., Chen, Y., Sun, S., Ma, C., & Song, M. (2021). DEAL: Difficulty-Aware Active Learning for Semantic Segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12622 LNCS*, 672–688. https://doi.org/10.1007/978-3-030-69525-5_40 (cit. on p. 25)

Xu, H., & Mannor, S. (2010). Robustness and generalization. https://doi.org/10.48550/ARXIV.1005.2243. (Cit. on p. 13)

Yang, Y., Ma, Z., Nie, F., Chang, X., & Hauptmann, A. G. (2015). Multi-Class Active Learning by Uncertainty Sampling with Diversity Maximization. *International Journal of Computer Vision*, *113*(2), 113–127. https://doi.org/10.1007/s11263-014-0781-x (cit. on p. 7)

Yang, Y., & Newsam, S. (2010). Bag-of-visual-words and spatial extensions for land-use classification. *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS)* (cit. on p. 21).

Yao, H., Zhang, L., & Finn, C. (2021). Meta-learning with fewer tasks through task interpolation. https://doi.org/10.48550/ARXIV.2106.02695. (Cit. on p. 38)

Yap, P., Ritter, H., & Barber, D. (2020). Addressing catastrophic forgetting in few-shot problems. https://doi.org/10.48550/ARXIV.2005.00146. (Cit. on p. 43)

Yin, M., Tucker, G., Zhou, M., Levine, S., & Finn, C. (2019). Meta-Learning without Memorization. https://github.com/google-research/%20http://arxiv.org/abs/1912.03820 (cit. on p. 36)

Yoo, D., & Kweon, I. S. (2019). Learning loss for active learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 93–102. https://doi.org/10.1109/CVPR.2019.00018 (cit. on p. 11)

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., & Ahn, S. (2018). Bayesian model-agnostic meta-learning. *Advances in Neural Information Processing Systems*, *2018-Decem*, 7332–7342 (cit. on p. 37).

Zhang, H., Li, Y., Jiang, Y., Wang, P., Shen, Q., & Shen, C. (2019). Hyperspectral Classification Based on Lightweight 3-D-CNN with Transfer Learning. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(8), 5813–5828. https://doi.org/10.1109/TGRS.2019.2902568 (cit. on p. 2)

Zhang, Z., Pasolli, E., & Crawford, M. M. (2019). Crop Mapping through an Adaptive Multiview Active Learning Strategy. *2019 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, 307–311. https://doi.org/10.1109/MetroAgriFor.2019.8909253 (cit. on p. 26)

Zhdanov, F. (2019). Diverse mini-batch Active Learning. http://arxiv.org/abs/1901.05954 (cit. on pp. 8, 10)

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. https://doi.org/10.1109/JPROC.2020.3004555. (Cit. on p. 2)