Universitat Autònoma de Barcelona

Reducing Label Effort with Deep Active Learning

Javad Zolfaghari Bengar

Universitat Autònoma
de Barcelona

Reducing Label Effort with Deep Active Learning

A dissertation submitted by **Javad Zolfaghari Bengar** to the Universitat Autònoma de Barcelona in fulfilment of the degree of **Doctor of Philosophy** in the Departament de Ciències de la Computació.

Bellaterra, November 5, 2021

Director	**Dr. Joost van de Weijer** Computer Vision Centre Universitat Autònoma de Barcelona **Dr. Bogdan Raducanu** Computer Vision Centre Universitat Autònoma de Barcelona
Thesis committee	**Dr Jan van Gemert** Department of Intelligent Systems Delft University of Technology **Dr Javier Ruiz-Hidalgo** TSC (Signal Theory and Communications Department) Universitat Politecnica de Catalunya **Dr Lluis Gomez** Computer Vision Centre Universitat Autònoma de Barcelona

This document was typeset by the author using LaTeX 2ε.

ISBN: 978-84-122714-9-2

Printed by Ediciones Gráficas Rey, S.L.

Acknowledgements

I am extremely thankful to my supervisors Dr. Joost van de Weijer and Dr. Bogdan Raducanu for their noble guidance and encouragement. It wouldn't have been possible to conduct this research without their precious support. I'm proud of, and grateful for my time working with Joost and Bogdan. Immense gratitude to Joost for his knowledge, intellectuality and benevolence. I learned a lot from him as an inspiring team leader who gave me opportunities to engage in different research projects. Can't forget the time when he was proofreading the papers late at night and early in the morning. Sincere gratitude to Bogdan who guided me throughout my research work, for his knowledge and valuable suggestions, ever encouraging and motivating guidance and helping me settle down. All these supports have given me more power and spirit to excel in my research. They all really mean a lot to me.

I am pleased to thank my friends in the laboratory Marc Castelló Torrellas and Enric Sala Esteva whose company will always be remembered. I learned a lot from their insightful suggestions in many different aspects. I would like to thank Dr. Abel Gonzalez as my post-doc for the fruitful discussions and for the days we were working together before the deadlines. I would like to acknowledge Dr. Marc Massana for providing me with his guidance regarding computational resources. I would like to thank all the friends and colleagues in the LAMP group and CVC who made the time of my PhD pleasurable and memorable. Many thanks to CVC staff, Montse and Gigi and others who helped me in one way or another. Also Many thanks are given to the committee members who reviewed this thesis.

Last but not the least, I am grateful for my parents whose constant love and support keep me motivated and confident from thousands kilometers away. Deepest thanks to my brothers, who keep me grounded, remind me of what is important in life, and are always supportive of my adventures. I am forever thankful for the unconditional love and support throughout the entire thesis process and every day.

Abstract

Deep convolutional neural networks (CNNs) have achieved superior performance in many visual recognition applications, such as image classification, detection and segmentation. Training deep CNNs requires huge amounts of labeled data, which is expensive and labor intensive to collect. Active learning is a paradigm aimed at reducing the annotation effort by training the model on actively selected informative and/or representative samples. In this thesis we study several aspects of active learning including video object detection for autonomous driving systems, image classification on balanced and imbalanced datasets and the incorporation of self-supervised learning in active learning. We briefly describe our approach in each of these areas to reduce the labeling effort.

In chapter two we introduce a novel active learning approach for object detection in videos by exploiting temporal coherence. Our criterion is based on the estimated number of errors in terms of false positives and false negatives. Additionally, we introduce a synthetic video dataset, called SYNTHIA-AL, specially designed to evaluate active learning for video object detection in road scenes. Finally, we show that our approach outperforms active learning baselines tested on two outdoor datasets.

In the next chapter we address the well-known problem of over confidence in the neural networks. As an alternative to network confidence, we propose a new informativeness-based active learning method that captures the learning dynamics of neural network with a metric called label-dispersion. This metric is low when the network consistently assigns the same label to the sample during the course of training and high when the assigned label changes frequently. We show that label-dispersion is a promising predictor of the uncertainty of the network, and show on two benchmark datasets that an active learning algorithm based on label-dispersion obtains excellent results.

In chapter four, we tackle the problem of sampling bias in active learning methods on imbalanced datasets. Active learning is generally studied on balanced datasets where an equal amount of images per class is available. However, real-world datasets suffer from severe imbalanced classes, the so called long-tail distribution. We argue that this further complicates the active learning process, since the

imbalanced data pool can result in suboptimal classifiers. To address this problem in the context of active learning, we propose a general optimization framework that explicitly takes class-balancing into account. Results on three datasets show that the method is general (it can be combined with most existing active learning algorithms) and can be effectively applied to boost the performance of both informative and representative-based active learning methods. In addition, we show that also on balanced datasets our method generally results in a performance gain.

Another paradigm to reduce the annotation effort is self-training that learns from a large amount of unlabeled data in an unsupervised way and fine-tunes on few labeled samples. Recent advancements in self-training have achieved very impressive results rivaling supervised learning on some datasets. In the last chapter we focus on whether active learning and self supervised learning can benefit from each other. We study object recognition datasets with several labeling budgets for the evaluations. Our experiments reveal that self-training is remarkably more efficient than active learning at reducing the labeling effort, that for a low labeling budget, active learning offers no benefit to self-training, and finally that the combination of active learning and self-training is fruitful when the labeling budget is high.

Key words: *visual recognition, deep active learning, video object detection, semi-supervised learning, imbalance datasets, self-supervised learning*

Resumen

Las redes neuronales convolucionales profundas (CNNs) han logrado un rendimiento superior en muchas aplicaciones de reconocimiento visual, como la clasificación, detección y segmentación de imágenes. El entrenamiento de CNNs profundas requiere grandes cantidades de datos etiquetados, que tienen un alto coste y son laboriosos de conseguir. El aprendizaje activo es un paradigma destinado a reducir el esfuerzo de anotación entrenando el modelo con muestras informativas y / o representativas seleccionadas de una manera activa. En esta tesis estudiamos varios aspectos del aprendizaje activo, incluida la detección de objetos de video para sistemas de conducción autónoma, la clasificación de imágenes en conjuntos de datos balanceados y no balanceados y la incorporación del aprendizaje auto-supervisado en el aprendizaje activo. Describimos brevemente nuestro enfoque en cada una de estas áreas para reducir el esfuerzo de etiquetado.

En el capítulo dos presentamos un nuevo enfoque de aprendizaje activo para la detección de objetos en videos haciendo uso de la coherencia temporal. Nuestro criterio se basa en el número estimado de errores en términos de falsos positivos y falsos negativos. Además, presentamos un conjunto de datos de video sintético, llamado SYNTHIA-AL, especialmente diseñado para evaluar el aprendizaje activo para la detección de objetos de video en escenas de la carretera. Finalmente, mostramos que nuestro método supera unos métodos de referencia de aprendizaje activo probadas en dos conjuntos de datos en adquiridos en el exterior.

En el próximo capítulo abordamos el conocido problema de la sobre confianza en las redes neuronales. Como alternativa a la confianza en la red, proponemos un nuevo método de aprendizaje activo basado en un criterio informativo que captura la dinámica de aprendizaje de la red neuronal con una métrica llamada dispersión de etiquetas. Esta métrica es baja cuando la red asigna constantemente la misma etiqueta a la muestra durante el proceso entrenamiento y alta cuando la etiqueta asignada cambia con frecuencia. Mostramos que la dispersión de etiquetas es un predictor prometedor de la incertidumbre de la red, y mostramos en dos conjuntos de datos de referencia que un algoritmo de aprendizaje activo basado en

la dispersión de etiquetas obtiene excelentes resultados.

En el capítulo cuatro, abordamos el problema del sesgo de muestreo en los métodos de aprendizaje activo sobre conjuntos de datos no balanceados. El aprendizaje activo se estudia generalmente en conjuntos de datos balanceados donde se encuentra disponible la misma cantidad de imágenes por clase. Sin embargo, los conjuntos de datos del mundo real consisten de clases severamente no balanceadas, la denominada distribución de cola larga. Argumentamos que esto complica aún más el proceso de aprendizaje activo, ya que el conjunto de datos no balanceado puede dar lugar a clasificadores subóptimos. Para abordar este problema en el contexto del aprendizaje activo, proponemos un marco de optimización general que tiene en cuenta explícitamente el balance de las clases. Los resultados en tres conjuntos de datos muestran que el método es general (se puede combinar con la mayoría de los algoritmos de aprendizaje activo existentes) y se puede aplicar de manera efectiva para impulsar el rendimiento de los métodos de aprendizaje activo tanto informativos como representativos. Además, demostramos que también en conjuntos de datos balanceados, nuestro método, en general, mejora el rendimiento.

Otro paradigma para reducir el esfuerzo de anotación es el aprendizaje auto-supervisado que aprende de una gran cantidad de datos sin etiquetar de una manera no supervisada y se ajusta con pocas muestras etiquetadas. Los avances recientes en el aprendizaje auto-supervisado han logrado resultados muy impresionantes que rivalizan con el aprendizaje supervisado en algunos conjuntos de datos. En el último capítulo nos enfocamos en si el aprendizaje activo y el aprendizaje auto-supervisado pueden beneficiarse mutuamente. Sobre los conjuntos de datos para el reconocimiento de objetos, estudiamos con conjuntos de datos etiquetados de distintos tamaños para las evaluaciones. Nuestros experimentos revelan que el aprendizaje auto-supervisado es notablemente más eficiente que el aprendizaje activo para reducir el esfuerzo de etiquetado, que para un presupuesto de etiquetado bajo, el aprendizaje activo no ofrece ningún beneficio para el auto-aprendizaje y, finalmente, que la combinación de aprendizaje activo y auto-supervisado es útil cuando el presupuesto de etiquetado es elevado.

Palabras clave: *reconocimiento visual, aprendizaje activo profundo, detección de objetos en video, aprendizaje semi-supervisado, conjuntos de datos imbalanceados, aprendizaje auto-supervisado*

Resum

Les xarxes neuronals convolucionals profundes (CNN) han aconseguit un rendiment superior en moltes aplicacions de reconeixement visual, com la classificació, detecció i segmentació d'imatges. El entrenament de CNN profundes requereix grans quantitats de dades etiquetades, que tenen un alt cost i son laboriosos de recollir. L'aprenentatge actiu és un paradigma dirigit a reduir l'esforç d'anotació entrenant el model en mostres informatives i/o representatives seleccionades d'una manera activa. En aquesta tesi estudiem diversos aspectes de l'aprenentatge actiu, com ara la detecció d'objectes de vídeo per a sistemes de conducció autònoma, la classificació d'imatges en conjunts de dades balancejats i no balancejats i la incorporació de l'aprenentatge auto-supervisat en l'aprenentatge actiu. Descrivim breument el nostre enfocament en cadascuna d'aquestes àrees per reduir l'esforç d'etiquetatge.

Al capítol dos introduïm un nou enfocament d'aprenentatge actiu per a la detecció d'objectes en vídeos aprofitant la coherència temporal. El nostre criteri es basa en el nombre estimat d'errors en termes de falsos positius i falsos negatius. A més, introduïm un conjunt de dades de vídeo sintètic, anomenat SYNTHIA-AL, especialment dissenyat per avaluar l'aprenentatge actiu per a la detecció d'objectes de vídeo en escenes de carretera. Finalment, mostrem que el nostre enfocament supera les línies de base d'aprenentatge actiu provades en dos conjunts de dades a l'exterior.

En el següent capítol abordem el conegut problema de sobre confiança en les xarxes neuronals. Com a alternativa a la confiança en xarxa, proposem un nou mètode d'aprenentatge actiu basat en un criteri informatiu que captura la dinàmica d'aprenentatge de la xarxa neuronal amb una mètrica anomenada dispersió d'etiquetes. Aquesta mètrica és baixa quan la xarxa assigna constantment la mateixa etiqueta a la mostra durant el procés d'entrenament i alta quan l'etiqueta assignada canvia amb freqüència. Mostrem que la dispersió d'etiquetes és un predictor prometedor de la incertesa de la xarxa i mostrem en dos conjunts de dades de referència que un algorisme d'aprenentatge actiu basat en la dispersió d'etiquetes obté resultats excel·lents.

Al capítol quatre, abordem el problema del biaix de mostreig en mètodes d'a-

prenentatge actiu sobre conjunts de dades no balancejats. L'aprenentatge actiu s'estudia generalment en conjunts de dades balancejats on hi ha disponible la mateixa quantitat d'imatges per classe. Tanmateix, els conjunts de dades del món real consisteixen de classes severament no balancejats, l'anomenada distribució de cua llarga. Argumentem que això complica encara més el procés d'aprenentatge actiu, ja que el conjunt de dades no balancejats pot donar lloc a classificadors subòptims. Per abordar aquest problema en el context de l'aprenentatge actiu, proposem un marc d'optimització general que tingui en compte explícitament el balanç de classe. Els resultats de tres conjunts de dades van mostrar que el mètode és general (es pot combinar amb la majoria dels algorismes d'aprenentatge actiu existents) i es pot aplicar de manera eficaç per augmentar el rendiment dels mètodes d'aprenentatge actiu tant informatius com representatius. A més, demonstrem que també en conjunts de dades balancejats, el nostre mètode, en general, millora el rendiment.

Un altre paradigma per reduir l'esforç d'anotació és l'aprenentatge auto-supervisat que aprèn d'una gran quantitat de dades sense etiquetar de manera no supervisada i afina en poques mostres etiquetades. Els avenços recents en l'aprenentatge auto-supervisat han aconseguit resultats molt impressionants que rivalitzen amb l'aprenentatge supervisat en alguns conjunts de dades. En el darrer capítol ens centrem en si l'aprenentatge actiu i l'aprenentatge auto-supervisat es poden beneficiar mútuament. Sobre els conjunts de dades per al reconeixement d'objectes, estudiem amb conjunts de dades etiquetades de diferents mides per a les avaluacions. Els nostres experiments revelen que l'aprenentatge auto-supervisat és notablement més eficient que l'aprenentatge actiu per reduir l'esforç d'etiquetatge, que per a un baix pressupost d'etiquetatge, l'aprenentatge actiu no ofereix cap benefici per a l'aprenentatge auto-supervisat i, finalment, la combinació d'aprenentatge actiu i auto-supervisat és útil quan el pressupost d'etiquetatge és elevat.

Paraules clau: *reconeixement visual, aprenentatge actiu profund, detecció d'objectes en vídeo, aprenentatge semi-supervisat, conjunts de dades no balancejats, aprenentatge auto-supervisat*

Contents

List of Figures

List of Tables

1 Introduction

Machine learning technology facilitated many aspects of modern life: web searches, products like cameras and smartphones, recommendations systems in e-commerce platforms and autonomous systems in industries such as car manufacturing. They are used to recognize objects in images, transcribe speech into text, match posts or products with users' interests, and find relevant results of search. Deep learning is a class of techniques that is widely used in these applications. It is very promising across various modalities including image, text, and speech recognition.

The most common form of machine learning, in either deep or classical frameworks, is supervised learning. In computer vision, to build a system that can classify images as containing, say, a house, a car, a person or a pet we first need to collect a large dataset of images of houses, cars, people and pets, each labeled with its category. During training, the system is shown an image and produces an output in the form of a vector of scores, one for each category. We want the desired category to have the highest score of all categories. Deep learning is resource hungry and requires specially large amounts of data. In a typical deep learning system, there may be hundreds of millions of labeled examples with which to train the machine.

However, labeled examples are often difficult to obtain. For example, in the object detection task, the annotation include drawing a bounding box around every object in the image together with a class label that the object belongs to. Active learning systems attempt to overcome the labeling bottleneck by asking queries in the form of unlabeled samples to be labeled by an oracle (e.g., a human annotator). In this way, the active learner aims to achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data. Active learners may query samples in several scenarios and various strategies to decide which samples are most informative. In the context of computer vision, this thesis studies strategies and scenarios in using pool-based active learning where the queries are selected from a large pool of unlabeled images or frames in the case of video.

1.1 Active Learning

Active learning is a subfield of machine learning and artificial intelligence in general. The main hypothesis is that, if the learning algorithm is allowed to choose the data to train on, it will perform better with less training. This is a desirable property for a learning algorithm because in many supervised learning systems in order for the algorithm to work well, it must be trained often with millions of labeled samples. Although sometimes, the labels come at little or no cost such as "spam" flag you mark on unwanted email. These flags will be used by the algorithm to better filter junk emails. In this case the information you provide is for free but for many advanced supervised learning systems labels are very difficult, time-consuming, or expensive to obtain.

In an active learning framework, a learner may begin with a small number of instances in the labeled training set, request labels from the pool of unlabeled data for one or more carefully selected instances, learn from the query results, and then leverage its new knowledge to choose which instances to query next. Once a query has been made, there are usually no additional assumptions on the part of the learning algorithm. The new labeled instance is simply added to the labeled set, and the learner proceeds from there in a standard supervised way. Figure 1.1 illustrates active learning framework.

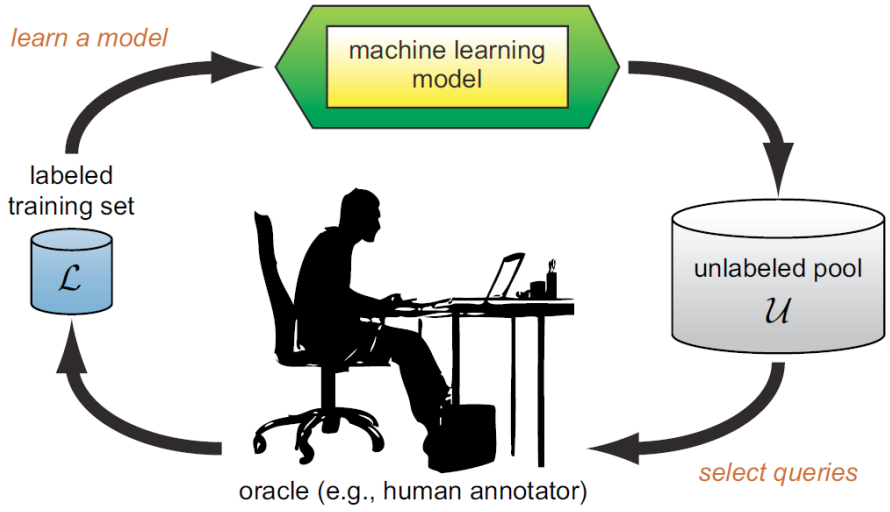
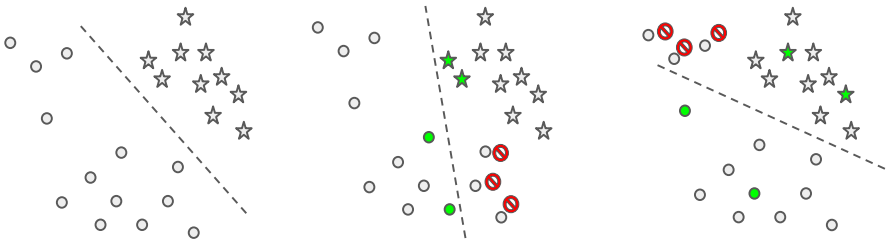


Figure 1.1 – **Pool based active learning cycle [116]**



(a) Binary classification problem (b) Informative queries (c) Representative queries

Figure 1.2 – **An illustrative example for selecting informative and representative instances** Notice that the circles and stars represent two different classes, green points are queried samples using different methods and dashed line represent the classifier trained on queried samples. It can be seen the models in (b) and (c) fail to classify all the instances correctly.

Generally speaking, there are two main sampling criteria in designing an effective active learning algorithm, that is, informativeness and representativeness [62]. Informativeness represents the ability of a sample to reduce the generalization error of the adopted classification model, and ensures less uncertainty of the classification model in the next cycle. Informativeness methods query samples that are either uncertain or dissimilar to labelled samples, regardless of relationship to other unlabelled samples using posterior probability. Representativeness decides whether a sample can exploit the structure underlying unlabeled data. These methods query samples that represent well the underlying distribution of the input data to maximize the benefit of labelling and prevent querying uninformative outliers. Fig. 1.2 illustrates an example of instances quiered by each method.

Deep learning is limited by the high cost of labeling. In comparison, an effective AL algorithm can theoretically achieve exponential acceleration in labeling efficiency [4, 106]. This huge potential saving in labeling costs is a fascinating development. However, the classic AL algorithm also finds it difficult to handle high-dimensional data [122]. Therefore, the combination of deep learning and active learning is expected to achieve superior results.

Based on an analysis of active learning literature we have identified four research directions which we have pursued in this thesis, and which we outline in the following subsections.

1.1.1 Active Learning for Object Detection in Video Sequences

Active learning is widely studied in image classification while only few works have investigated active learning for object detection, even though the problem of active learning is more pertinent for object detection than for image classification since the labeling effort also includes the more expensive annotation of the bounding box. One important application domain for which Object detection in sequences is of key importance is autonomous driving. It has noticeably improved partially due to the presence of large datasets. Further improvement however requires the collection of larger labeled datasets, which is both time and labor expensive.

In chapter 2 we focus on active learning for object detection in videos within an autonomous driving context. Ever since the introduction of the large-scale video object detection challenge ImageNet-VID [111], object detection in videos received more attention. The task is highly challenging due to phenomena such as detector flicker, i.e. the predicted outputs are severely affected by small changes in the input images. As a result many video-specific approaches are developed that require full video annotation. However, annotating all object instances in every frame is extremely costly. Hence recent datasets for autonomous driving provide a small subset of frames with object ground-truth annotations.

The inherent property of videos is *temporal coherence*, i.e. nearby frames usually contain the same instances in nearby locations. Fig. 1.3 shows eight consecutive frames of ILSRVC video dataset that contain two people riding bikes across in all the frames. This property can potentially be exploited to identify frames in which the detector might have wrongly detected objects (there is no support in nearby frames) or frames in which the detector failed to detect an object (there is evidence of the object in the surrounding frames). These frames are expected to be more beneficial to annotate than others, leading to potentially more accurate models when used for training.

Another issue in active learning for video object detection is that most active learning methods [44, 115, 116] are evaluated on simple image classification datasets such as MNIST [81] or CIFAR [78]. Approaches specific for object detection [15, 110, 126, 142] mainly use PASCAL VOC [38], covering various scene types. In the context of autonomous driving, only [110] uses a dataset depicting road scenes, KITTI [46]. Similarly to several other image datasets for autonomous driving [26, 143], KITTI is manually curated to contain mainly relevant information usable to train object detection models. This process is performed by human annotators who select interesting data samples containing cars, pedestrians, etc. The goal of active learning, however, is automatizing this process, making existing datasets not suitable for a proper evaluation since they lack the inherent redundancy present in automatically collected data. Ideally, a good dataset for evaluating

Figure 1.3 – From left to right shows eight consecutive frames of ILSVRC2015 VID dataset.

active learning contains a more raw version of the data, in which the image distribution is biased towards the uninteresting (e.g. empty road scenes) and highly redundant. Such dataset would better represent the type of data collected in a real setting, for example, video captured from a driving car. *Consequently, in this thesis we collect a more realistic dataset for active learning which has levels of redundancy closer to real-world applications. In addition, we aim to explore the usage of temporal coherence for the purpose of active learning.*

1.1.2 Active Learning based on Neural Network Dynamics

Exploiting network's uncertainty is one of the main approaches for active learning, as contained in its prediction, to select data for labeling. It is shown that neural networks sometimes make wrong predictions with high certainty [99]; overly confident with the predictions.

In a recent work, Toneva et al. [121] studied the learning dynamics during the training process of a neural network. For each training sample they track the transitions from being classified correctly to incorrectly (or vice-versa) over the course of learning. Based on these learning dynamics, they characterize a sample of being 'forgettable' (if its class label changes from subsequent presentation) or 'unforgettable' (if the class label assigned is consistent during subsequent presentations). Fig. 1.4 shows examples of forgettable and unforgettable samples.

They have shown that the unforgettable samples can be removed from training set without significant performance drop. However, the proposed method is not applicable to active learning as it requires labels to identify the un/forgettable samples. *Inspired by this work we aim to introduce a novel method to measure the uncertainty in model prediction by exploiting network dynamics. This measure can then be used to query samples that are informative for the model if labeled.*

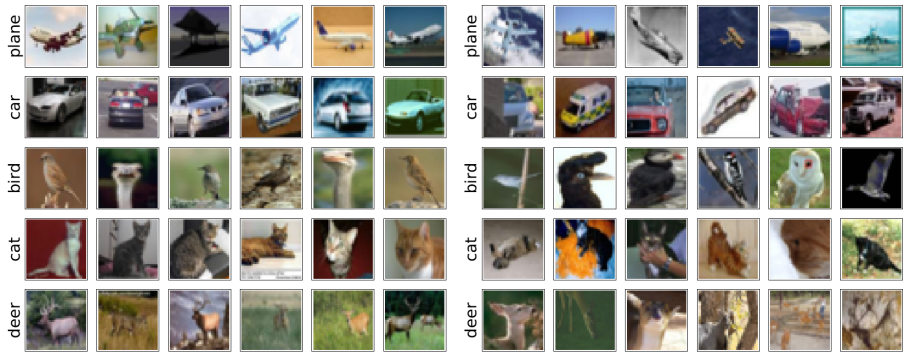


Figure 1.4 – Pictures of the most unforgettable (Left) and forgettable examples (Right) of five CIFAR-10 classes, when examples are sorted by number of forgetting events (ties are broken randomly). Forgettable examples seem to exhibit peculiar or uncommon features. Images are taken from [121].

1.1.3 Active Learning for Imbalanced Datasets

Visual recognition datasets in computer vision research are often almost uniformly distributed (e.g. CIFAR [77] and ILSVRC [79]). However, for many real-world problems data follows a long-tail distribution [97], meaning that a small number of head-classes are much more common than a large number of tail-classes (e.g. iNaturalist [124], landmarks [98]). See Fig. 1.5 for long-tail distribution of iNaturalist 2018 dataset.

Classification on such imbalanced datasets is an important research topic [28, 60, 105]. However, active learning is mostly studied on curated close-to-uniform datasets. Existing methods, regardless of how they are formulated, have a common underlying assumption that all classes are equal. They do not consider that some some classes might be more prevalent in the dataset than others. Instead, they focus on, given a data sample, how much error a trained model is expected to make, or the estimated uncertainties.

Closely related to the class-imbalance dataset problem, is the sampling bias problem which is a well-documented drawback of active learning [30, 92]. Datasets collected by active learning algorithms break the assumption that the data is identically and independently distributed (i.i.d), since the active learning algorithm might be biased towards particular regions of the unlabeled data manifold. One possible consequence of the sampling bias can be that the distribution over the classes no longer follow that of the unlabeled data pool. Several papers have investigated this

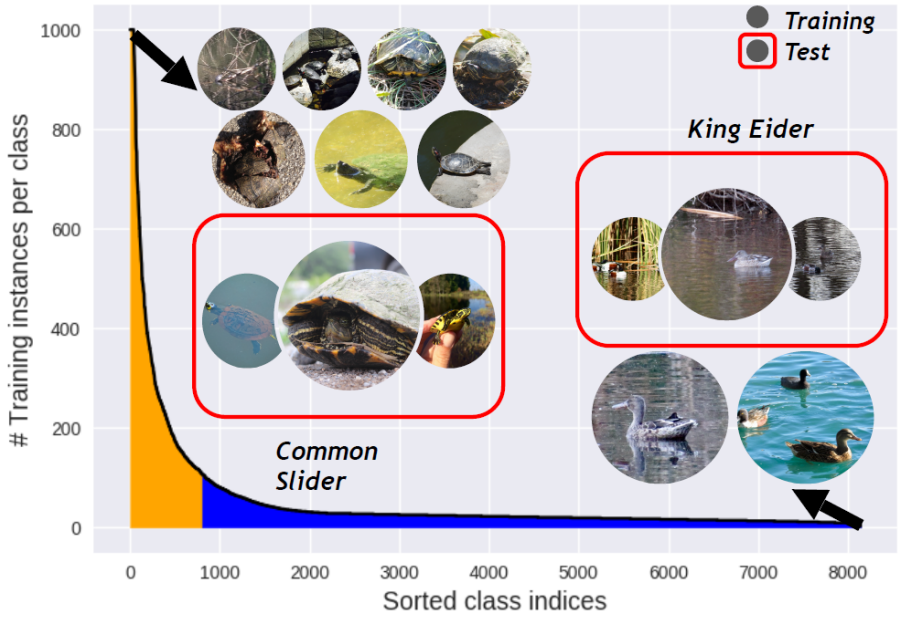


Figure 1.5 – The training set of iNaturalist 2018 exhibits a long-tailed class distribution. Image is taken from [64].

aspect of active learning however it remains not fully understood [12, 39].

Given the predominance of long-tail distributions, especially for real-world scenarios in which active learning is a crucial capability, we study active learning for imbalanced datasets. The aim is to minimize the labeling effort, while maximizing the performance when measured on a balanced test set.

1.1.4 Contribution of Self Supervised Learning in Active Learning

Active learning methods are typically evaluated by supervised training of the network on only the labeled data pool: the active learning method that obtains the best results, after a number of training cycles with a fixed label budget, is then considered superior.

Self-supervised learning of representation for visual data has seen stunning progress in recent years [19, 20, 21, 48, 55], with some unsupervised methods being able to learn representations that rival those learned supervised. The main progress has come from a recent set of works that learn representations that are invariant

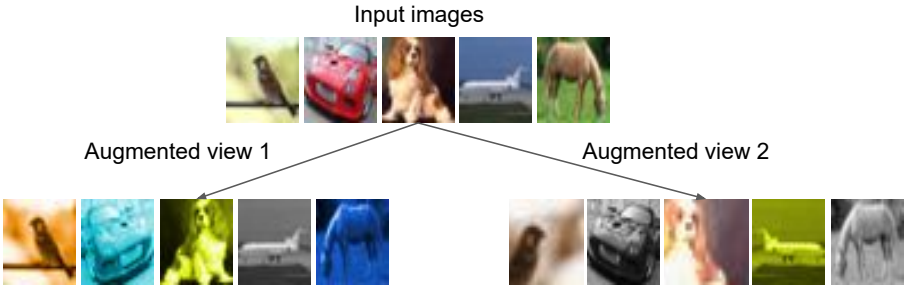


Figure 1.6 – Augmented views including cropping, flipping, blur and color distortion.

with respect to a set of distortions of the input data (such as cropping, applying blur, flipping, etc). See Fig.1.6 for the augmented views.

In these methods, two distorted views of the image are produced. Then the network is trained by enforcing the representations of the two views to be similar. To prevent these networks to converge to a trivial solution different approaches have been developed [48, 145]. The resulting representations are closing the gap with supervised-learned representation. For some downstream applications, such as segmentation and detection, the self-supervised representations even outperform the supervised representations [149].

Given the huge performance gains that are reported by applying self-supervised learning, we propose to re-evaluate existing active learning algorithms in this new setting where the unlabeled data is exploited by employing self-supervised learning.

1.2 Objectives and Approach

In the following we outline the objectives and approach for the four research lines that we have identified in the previous section:

1.2.1 Active Learning for Object Detection in Video Sequences

As mentioned active learning has mainly focused on image classification and there are relatively few works that focus on active learning for object detection in video sequences. We arrive at the following objective:

> **Temporal coherence for active learning:** Propose an active learning method that exploits the temporal coherence of video sequences. In addition, col-

lect a new dataset for active learning for object recognition in videos.

To reduce the annotation effort in the task of object detection in videos we propose to mainly annotate frames that contain detection errors. We estimate the number of errors in the frames by running the trained model on unlabeled frames. We consider two types of errors, false positives and false negatives, and show the effect of selecting either type. This exploratory experiment suggests a potentially powerful approach for active learning. Motivated by this, we develop a novel method to estimate detection errors in videos by exploiting the temporal coherence in the videos. We track detections forward and backward and define a graph on the detections that are temporally linked. Minimization of an energy function defined on this graphical model provides us with the detection of false positives and false negatives. These we subsequently use to select the frames to be annotated.

We also propose a new synthetic dataset specially designed for active learning in road scene videos. we have created a new synthetic dataset to evaluate active learning for object detection in road scenes. In particular, we modified the SYNTHIA environment [108] to generate the SYNTHIA-AL dataset using Unity Pro game engine. The aim is having an unbalanced foreground/background distribution, simulating the real collection scenario of a driving car. To make the dataset favorable for active learning a set of object classes and conditions are predominantly present, while other classes and conditions must appear less frequent. The data is generated by driving a car in a virtual world consisting of three different areas, namely town, city, and highway that are populated with a variety of pedestrians, cars, cyclists, and wheelchairs.

1.2.2 Active Learning based on Neural Network Dynamics

As discussed before the dynamics of labels for samples during the training of a neural network could be used to derive an uncertainty measure for each sample. We define therefore the following objective:

> **Network dynamics for active learning:** Propose an active learning method based on the learning dynamics of samples during the training of neural networks. This measure should address the overconfidence that is present in infromativeness methods that are directly based on network predictions.

In the context of active learning we propose a new uncertainty estimation method which is based on the learning dynamics of a neural network. With learning dynamics, we refer to the variations in the predictions of the neural network during training. Specifically, we keep track of the model predictions on every unlabeled sample dur-

ing the various epochs of training. Based on the variations of the predicted label of samples, we propose a new active learning metric called *label-dispersion*. This way, we can indirectly estimate the uncertainty of the model based on the unlabeled samples. We will directly use this metric as the acquisition function to select the samples to be labeled in the active learning cycles. Other than the forgetfulness measure proposed in [121], we do not require any label information.

1.2.3 Active Learning for Imbalanced Datasets

Imbalanced data are prevalent in many real-world applications. We therefore pursue the following objective:

> **Active learning for imbalanced data:** Study active learning method for imbalanced datasets. We aim to propose a method to address the data imbalance if present in the unlabeled data-pool.

To mitigate the problems caused by the sampling bias and imbalanced datasets, we introduce an optimization framework which corrects the class-imbalance presented in the unlabeled data pool, and aims to bias instead our selected samples to resemble the uniform distribution of the test set. Since we have no access to the class labels of the unlabeled data, we propose to trust the predicted labels, and use them to select a set of class-balanced images. This combination leads to a minimization problem, which can be formalized as a binary programming problem. We show that our optimization scheme is efficient, boosting the performance of both informativeness and representativeness methods.

1.2.4 Contribution of Self Supervised Learning in Active Learning

Self-supervised learning has seen considerable progress in recent years. Based on this observation, we propose the following objective:

> **Combining self-supervised learning and active learning:** Study the interaction between self-supervised learning and active learning; provide an analysis on how these two paradigms can benefit each other.

Self-supervised learning can learn high-quality features that are almost at par with the features learned by supervised methods. As such it has greatly improved the usefulness of unlabeled data. The standard active learning paradigm trains an algorithm on the labeled data set, and based on the resulting algorithm selects data points that are expected to be most informative for the algorithm in better

understanding the problem [117]. In this standard setup, the unlabeled data is not exploited to improve the algorithm. We perform extensive experiments on various datasets in terms of number of unlabeled data and categorise to study the contribution of self supervised methods in active learning methods. We analyze active learning and self supervised approaches independently and unified to investigate how they can benefit from each other.

2 Temporal Coherence for Active Learning in Videos*

Summary: *Autonomous driving systems require huge amounts of data to train. Manual annotation of this data is time-consuming and prohibitively expensive since it involves human resources. Therefore, active learning emerged as an alternative to ease this effort and to make data annotation more manageable. In this chapter, we introduce a novel active learning approach for object detection in videos by exploiting temporal coherence. Our active learning criterion is based on the estimated number of errors in terms of false positives and false negatives. The detections obtained by the object detector are used to define the nodes of a graph and tracked forward and backward to temporally link the nodes. Minimizing an energy function defined on this graphical model provides estimates of both false positives and false negatives. Additionally, we introduce a synthetic video dataset, called SYNTHIA-AL, specially designed to evaluate active learning for video object detection in road scenes. Finally, we show that our approach outperforms active learning baselines tested on two datasets.*

2.1 Introduction

For autonomous driving systems, the quality of object detection is of key importance. Its progress in recent years has been notable, partially due to the presence of large datasets [46, 143]. However, pushing detectors to further improve and finally be close to flawless, requires the collection of ever larger labeled datasets, which is both time and labor expensive. Active learning methods [116] tackle this problem by reducing the required annotation effort. The key idea behind active learning is that a machine learning model can achieve a satisfactory performance with a subset of the training samples if it is allowed to choose which samples to label. This contrasts with passive learning, where the data to be labeled is taken at random without taking into account the potential benefit of annotating each sample.

Active learning has been mainly investigated for the image classification task

*This chapter is based on a publication in the IEEE/CVF International Conference of Computer Vision Workshops, 2019 [7]

[31, 45, 68, 84, 86, 113, 136]. Only few works have investigated active learning for object detection, even though the problem of active learning is more pertinent for object detection than for image classification since the labelling effort also includes the more expensive annotation of the bounding box [73]. For instance, in [126, 141] the object detector is learned interactively in an incremental manner using a simple margin approach to select the most uncertain images. In [110], the active learning approach is based on a 'query-by-committee' strategy.

In this work we focus on active learning for object detection in videos. To the best of our knowledge, we are the first to consider this scenario. Object detection in videos has become of great interest ever since the introduction of the large-scale video object detection challenge ImageNet-VID [111]. The task has proven highly challenging due to phenomena such as detector flicker [67, 109], i.e. the drastic effects in the predicted outputs given by small changes in the images. This has spawn a multitude of video-specific approaches [70, 71, 129, 152, 153] that require comprehensive video annotation. However, exhaustively annotating all object instances in every frame is extremely costly. Possibly because of this, recent datasets for autonomous driving [93, 143] only offer a small subset of frames with object ground-truth annotations.

Video data has the inherent property of *temporal coherence*, i.e. nearby frames are expected to contain the same instances in nearby locations. This property can be exploited to identify frames in which the detector might have wrongly detected objects (there is no support in nearby frames) or frames in which the detector failed to detect an object (there is evidence of the object in the surrounding frames). These frames are expected to be more beneficial to annotate than others, leading to potentially more accurate models when used for training.

In this chapter, we confirm that annotating those frames that contain detection errors leads to higher accuracy given a limited annotation budget. We consider two types of errors, false positives and false negatives, and show the effect of selecting either type. This exploratory experiment suggests a potentially powerful approach for active learning. Motivated by this, we develop a novel method to estimate detection errors in videos by exploiting the temporal coherence in the videos. We track detections forward and backward and define a graph on the detections that are temporally linked. Minimization of an energy function defined on this graphical model provides us with the detection of false positives and false negatives. These we subsequently use to select the frames to be annotated. In summary, the contributions of this chapter are:

- We propose a new method for active learning in videos which exploits the temporal coherence.
- We propose a new synthetic dataset specially designed for active learning in

road scene videos.

- Our proposed method outperforms several baseline methods both on synthetic and real video data.

2.2 Related Work

Active learning for object detection. A critical aspect for an active learner is represented by the strategy used to query the next sample to be labeled. Four main query frameworks exist, which rely mostly on heuristics: informativeness [17, 44, 50, 139], representativeness [113, 115], hybrid [63, 138], and performance-based [43, 49, 114, 137]. Among all these, informativeness-based approaches are the most successful ones. A comprehensive survey of these frameworks and a detailed discussion can be found in [116]. Active learning has been successfully applied to a series of traditional computer vision tasks, such as image classification [45, 68, 72] (including medical image classification [113] and scene classification [86]), visual question answering (VQA) [88], image retrieval [147], remote sensing [31], action localization [58], and regression [42, 69].

With a strong emphasis on image classification, active learning for object detection has received less attention than expected due to the difficulty to aggregate several object hypothesis at frame level. Recently, [142] employed a loss module to learn the loss of a target model and select the images based on their output loss. However, in hybrid tasks such as object detection learning the loss is challenging. In [110], the active learning approach is based on a 'query-by-committee' strategy. A committee of classifiers is formed by the last convolutional layer of the base network together with the extra convolutional layers of the SSD architecture [90]. The disagreement between them for each candidate bounding box in an image is used as query strategy. In [126], the authors propose a system that learns object detectors on-the-fly, by refining its models via crowd-sourced annotations of web images. As active learning criterion, they use a simple margin approach which selects the most uncertain images which should be annotated. A similar idea is reported in [141], where an object detector is learned interactively, in an incremental manner. The system selects the images most likely to require user input based on an estimated annotation cost computed in terms of false positive and false negative detections. Other approaches to reduce the annotation cost for object detection are based on domain adaptation [59] or transfer learning [125].

In the current work, we introduce a novel active learning approach for object detection in videos, which exploits the temporal coherence of the found detections. The query strategy is based on the number of false positives and false negatives

detections identified using a graphical model.

Temporal coherence in video object detection. Several video object detection approaches [53, 70, 71, 89, 129, 152, 153] have attempted to use temporal information to enhance single-image object detectors [107] for multi-class video object detection. There are two main types of approaches.First, temporal information can be used to refine the detections output by the detector as a post-processing step. For example, Seq-NMS [53] re-scores detections using highly overlapping detections from surrounding frames. Some approaches [70, 71] are based on the concept of *tubelet*, i.e. spatio-temporal bounding boxes that span consecutive frames. T-CNN [71] uses tubelets, generated by tracking high confidence detections across frames, to re-score detections and recover false negatives.

The second type of approaches introduces temporal coherence while learning the features used by the model in an end-to-end manner. FGFA [152] uses optical flow to estimate the motion between frames, which is employed to learn features that aggregate information from surrounding frames, while [153] uses it for efficiency reasons, extracting features only for selected frames and propagating them to nearby frames. Contrary to the pixel-level approaches, Motion-Aware network [129] introduces instance-level feature aggregation by estimating the movement of proposals across frames and combining them. All these approaches use temporal information to improve object detection in videos, whereas we exploit it to select sets of samples in the context of active learning.

2.3 Active Learning for Video Object Detection

We describe here the general process of active learning applied to video object detection. Given a large pool of unlabeled data $\mathscr{D}_U$ (video frames) and an annotation budget b, the goal of active learning is to select a subset of b samples to be annotated as to maximize the performance of an object detection model (e.g. Faster R-CNN [107]). Active learning methods generally proceed sequentially by splitting the budget in several *cycles*. Here we consider the batch-mode variant [116], which annotates multiple samples per cycle, since this is the only feasible option for CNN training. At the beginning of each cycle, the model is trained on the labeled set of samples $\mathscr{D}_L$[†]. After training, the model is used to select a new set of samples to be annotated at the end of the cycle via an *acquisition function*. The selected samples are added to the labeled set $\mathscr{D}_L$ for the next cycle and the process is repeated until the annotation budget b is spent. Fig. 2.1 presents the active learning framework

[†]Most methods start with a small initial labeled set selected at random.

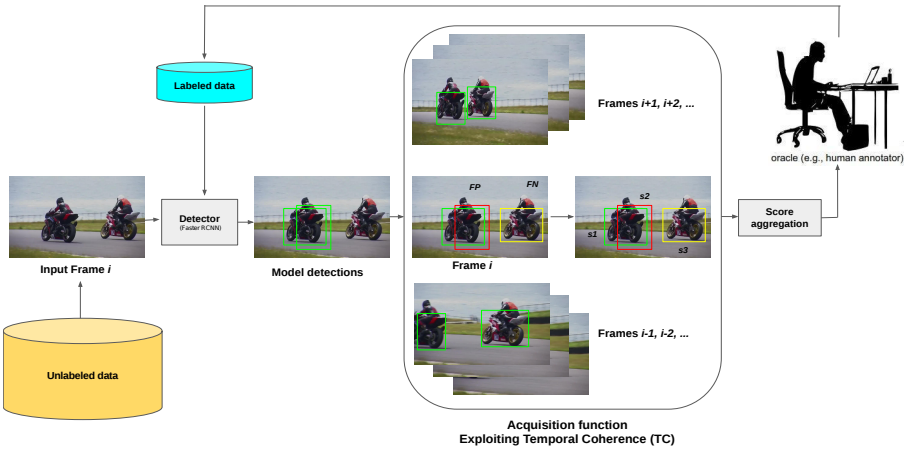


Figure 2.1 – **Overview of our active learning framework exploiting temporal coherence.** The detector outputs detections (green) for each frame in the unlabeled data. Considering the relationships between the detections of neighboring frames (both forward and backward), our temporal coherence acquisition function predicts false positive (red) and false negative (yellow) errors. Based on these predictions, each frame is given an aggregated score and ranked for selection. Finally the frames with top scores are annotated and added to the labeled data.

with our temporal coherence acquisition function, described in sec. 2.3.2. Note how each sample corresponds to an entire frame and thus all objects in the frame are annotated simultaneously.

In image classification, a single label is predicted per image (e.g. “the image contains a car”), and hence each data sample is a whole image. In object detection, however, the unit of prediction is a *region* inside the image, as the task also includes localization. This leads to the following open issue: should the annotation be performed for each individual region independently or considering all regions in the image simultaneously? Current deep learning algorithms for object detection [56, 107] consider fully annotated images during training, as negative examples are extracted from the unlabeled regions of positive images. Moreover, the most commonly used annotation tools for object detection [87, 112] employ image-wise annotations, requesting all regions for each presented image.This annotation style is more efficient as the cognitive burden of parsing the image can be shared across multiple regions. Therefore, we consider image-wise annotations in this chapter.

This is a more efficient approach since (i) parsing the image is a significant portion of the labelling effort, while annotating additional boxes only incur a small

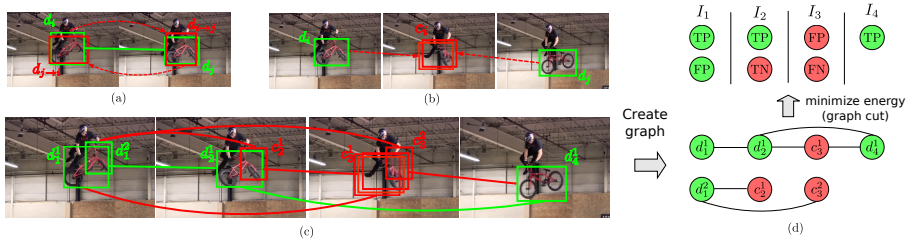


Figure 2.2 – **Error estimation using temporal coherence.** (a) Detections (green) across different frames are linked depending on the overlap with their corresponding tracks (red). (b) Candidate detections (red) are obtained by clustering tracked detections that do not overlap any local detection. (c) Example of detections, candidates, and their links for four consecutive frames. (d) Nodes of the generated graph using detections and candidates corresponding to figure (c). Once the graph is created, we minimize its energy via graph-cut to obtain and estimation of the errors in terms of FP and FN. In this example, we only track up to two surrounding frames, but in practice we use three.

extra cost, and (ii) some isolated regions are not very informative and thus they need to be presented in the context of the image nonetheless.(iii) most deep learning algorithms for detection consider fully labeled images (among others they consider the non labeled regions as negative examples during training). The former option is clearly inefficient as isolated regions are frequently not very informative, and thus the whole image must be presented. Once the annotator observes it, only requesting a particular region is rather wasteful, as annotating all potential remaining instances would only incur a small additional cost.

The acquisition function is the most crucial component and the main difference between active learning methods in the literature. In general, an acquisition function φ receives a sample x and outputs a score $\varphi(x)$ indicating how valuable x is for training the current model. More sophisticated acquisition functions may consider additional data such as the samples already selected for the current batch, the previously labeled samples $\mathscr{D}_L$, or the unlabeled pool $\mathscr{D}_U$ (see [116] for details). In the remainder of this section, we introduce our two proposed acquisition functions for video object detection in road scenes. Sec. 2.3.1 presents an exploratory function that approximates a performance upper bound. Sec. 2.3.2 describes our main contribution: a practical acquisition function based on temporal coherence and specialized for video object detection.

2.3.1 Oracle-based acquisition

The underlying assumption of active learning is that some data samples provide more valuable information than others, so that when labeled and used for training, they improve the model performance by decreasing the number of errors. A suitable acquisition function would select those samples in which the network commits the greatest number of errors so they can be remedied. Assuming perfect generalization from training to test data, such function would be an upper bound for all active learning methods.[‡] Motivated by this and in order to study the potential of active learning for video object detection, we propose here an *oracle-based* acquisition function to implement this desirable behavior.

Our oracle-based active selection uses ground-truth information to quantify the number of errors in a given image, and selects those images that have the greatest number of errors. Note this is not a useful active learning function in practice, as we would not have access to the ground-truth annotations in a real scenario. We consider two types of errors that directly affect the usually employed object detection metric of Average Precision (AP) [38, 87]: False Positives (FP) and False Negatives (FP). Let us consider a detection as *correct* if it overlaps a ground-truth bounding box more than 0.5, using the Intersection-over-Union (IoU) measure for overlap [38]. FPs are detections that are not correct (i.e. have little or no overlap with any ground-truth) or are duplicated, while FNs are those ground-truth instances that have not been detected. We consider two different acquisition functions, one which considers the number of FPs in a frame and the other which considers the number of FNs in a frame[§]. Since the acquisition scores of these functions are integer numbers, it is frequent to have ties between images. We disambiguate between ties by random selection.

2.3.2 Temporal coherence for error estimation

Video data has the inherent property of *temporal coherence*, i.e. nearby frames are expected to contain the same instances in nearby locations. Based on this, we propose a method to estimate the errors of a video object detector by exploiting the expected temporal coherence, and then use the estimates with the oracle-based acquisition function proposed in sec. 2.3.1, but using estimations as oracle. Let us consider a video v composed of a sequence of L frames $\{I_1, ..., I_L\}$. An object

[‡] In practice, a decrease in errors in the training set may not necessarily lead to better performance in a separate test set, making this acquisition function an *approximation* to the upper bound.

[§] We experimented with combining both FP and FN in the acquisition function but found this to not improve results.

detector outputs a set of detections $D_i = \{d_i^0, ..., d_i^K\}$ for each frame I_i[¶]. Temporal coherence induces a bijective mapping between sets of detections in nearby frames when corrected for minor localization changes. In order to correct such changes we employ an object tracker, of which details follow later. Formally, given a detection d_i^k in frame I_i, the tracker estimates the location of the contents of this region in frame I_j, which we refer to as $d_{i\to j}^k$. The tracking can be performed in the direction of time ($i < j$) or in the reverse direction. The set of all tracked detections $D_{i\to j} = \{d_{i\to j}\}$ can be thought of as weak detections obtained via temporal coherence using another frame's detections, rather than being directly predicted by the object detector based on the frame's content. We can now link detections of the same class across frames based on their tracked detections. More concretely, we link detection d_i^k in frame I_i with detection d_j^l in I_j if $\text{IoU}(d_i^k, d_{j\to i}^l) > \theta$ or $\text{IoU}(d_j^l, d_{i\to j}^k) > \theta$ (Fig. 2.2a). That is, if any of the tracked detections (forward or backward) overlaps the other detection in the corresponding frame. Note how there might be tracked detections that are not matched with any local detection (Fig. 2.2b). Such tracked detections could indicate the presence of an instance in that frame that has been missed by the detector. We cluster groups of unmatched tracked detections in the same frame based on their overlap. We term these groups as detection *candidates* and use the notation c_i^k for the k-th candidate of frame I_i. Each detection d_i can either be a True Positive (TP) if it correctly localizes an object instance in the image, or a FP if it erroneously predicts the presence of a particular object. On the other hand, a detection candidate c_i can be a True Negative (TN) if no object instance is present in its location, or a FN if it corresponds to a missed detection. We now estimate the type of every detection and detection candidate by formalizing our approach as a graphical model.

Graphical model. Let us express all detections and candidates as a set of binary random variables $\mathcal{V} = \{v_1, ..., v_N\}$, where $v_n = d$ if it corresponds to a detection d_i^k and $v_n = c$ for a candidate c_i^k. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with vertices $\mathcal{V}$ and edges $\mathcal{E}$ between connected detections across different frames (via the links previously introduced) and candidates connected with their originating detections (see Fig. 2.2). Each v_n can take one of four possible labels: TP, FP, TN, or FN. We consider the following energy function on label assignment $\mathcal{L}$:

$$E(\mathcal{L}) = \sum_{v\in\mathcal{V}} \phi_v(l_v) + \sum_{v_1,v_2\in\mathcal{C}} \psi_{v_1,v_2}(l_{v_1}, l_{v_2}), \tag{2.1}$$

[¶]Here we consider object detectors that process each frame independently, such as Faster R-CNN [107].

where $\phi_v(l_v)$ is the unary cost of assigning label l_v to v and $\psi_{v_1,v_2}(l_{v_1}, l_{v_2})$ is the pairwise cost of assigning the label pair (l_{v_1}, l_{v_2}) to a pair of connected variables $(v_1, v_2) \in \mathcal{E}$. We define the unary cost for detection variables as

$$\phi_{v=d}(l_v) = \begin{cases} 0 & \text{if } l_v = \text{TP} \\ \infty & \text{if } l_v = \text{TN} \\ 1 & \text{if } l_v = \text{FP} \\ \infty & \text{if } l_v = \text{FN} \end{cases} \tag{2.2}$$

This indicates that in principle we trust the outputs of the detector and that assigning a contradicting label should incur some cost. By definition, detections are 'positives' and thus assigning a 'negative' label is strongly discouraged. Analogously, the unary cost for candidate variables is

$$\phi_{v=c}(l_v) = \begin{cases} \infty & \text{if } l_v = \text{TP} \\ 0 & \text{if } l_v = \text{TN} \\ \infty & \text{if } l_v = \text{FP} \\ 1 & \text{if } l_v = \text{FN} \end{cases} \tag{2.3}$$

In this case, candidates can only be negatives as they are not part of the original outputs of the detector and hence cannot be positives.

We specify the pairwise cost using the following matrix

$$\psi_{v_1,v_2}(l_{v_1}, l_{v_2}) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \tag{2.4}$$

where the considered label assignment order is $l_v = (\text{TP, FP, TN, FN})$. This indicates that TP should be connected with other TP or FN, whereas FP are preferably connected with other FP or with TN. Intuitively, the pairwise cost enforces temporal coherence between the detections and the candidates, propagating the correctness to connected variables and collaboratively determining the errors.

We optimize the energy function in (2.1) via graph cut [76], which finds the globally optimal solution by solving the dual max-flow problem. In fact, the problem can be reduced to a binary labelling problem, considering only two possible labels (True or False) with different meanings depending on the type of input variable, i.e. positives for detections and negatives for candidates. We use the graph-cut implementation in the Python library PyMaxflow [14].

Acquisition function. Once all variables in $\mathcal{V}$ have been assigned their optimal labels, we record the estimated number of FPs and FNs contained in each frame. We revert now to the oracle-based acquisition function described in sec. 2.3.1, but using error estimates instead of actual errors, which makes the function is useful in practice as it does not require any ground-truth information. We refer to this acquisition function as Temporal Coherence (TC). Experimental results show similar performance when considering only FP, only FN, or both FP and FN. Therefore, we use only the number of FP for the acquisition function of TC.

Object tracker. In order to temporally link detections and construct connections between graph nodes, we considered two types of object trackers, namely Optical Flow (PWC-NET) [120] and SiamFC tracker [11]. To utilize optical flow for the purpose of object tracking, we first compute a dense 2D real-valued vector map of the motions between all pairs of consecutive frames in the dataset. Then, we translate the box coordinates using the motion vector corresponding to the box center to obtain the tracked box in the next or previous frame. As an alternative to track detections we employ SiamFC [11], a state of the art Siamese-based object tracker. The bottleneck of this tracking method in the context of active learning is that, despite its efficiency, it imposes a huge computational burden when tracking detections every cycle, given the vast amount of detections. On the contrary, optical flow is only computed once at the beginning and can be used throughout all cycles with a negligible overhead.

2.4 Synthetic Dataset

Most active learning methods [44, 115, 116] are evaluated on simple image classification datasets such as MNIST [81] or CIFAR [78]. Approaches specific for object detection [15, 110, 126, 142] mainly use PASCAL VOC [38], covering various scene types. In the context of autonomous driving, only [110] uses a dataset depicting road scenes, KITTI [46]. Similarly to several other image datasets for autonomous driving [26, 143], KITTI is manually curated to mostly contain relevant knowledge usable to train object detection models. This process is performed by human annotators that select interesting data samples containing cars, pedestrians, etc. The goal of active learning, however, is automatizing this process, making existing datasets not suitable for a proper evaluation. Ideally, a good dataset for evaluating active learning contains a more raw version of the data, in which the image distribution is unbalanced towards the uninteresting (e.g. empty road scenes) and highly redundant. Such dataset would better represent the type of data collected in a real setting, for example, video captured from a driving car. For this reason,

Subset Name	Seq.	Frames	Area	Conditions	P(Pe/Cy/Ca/Wh)
Default	150	74K	C,H	S,W,F,R	30/20/35/0
Town	36	17K	T	S,W,F,R	30/20/35/0
Night	6	3K	C,H	N	0/0/35/0
Wheelchair	5	2K	C,T	S	20/20/0/100
Test (no WC)	85	40K	C,H,T	S,F,R,N	30/20/35/0
Test (WC)	12	5K	C,T	S	20/20/0/100

Table 2.1 – **SYNTHIA-AL data distribution.** Seq. indicates the number of videos. Environment conditions are Fall (F), Winter (W), Spring (S), Rain (R), and Night (N). Areas are City (C), Town (T), and Highway (H). The spawning probabilities are given for pedestrians (Pe), cyclists (Cy), cars (Ca), and wheelchairs (Wh).

and following recent trends [108, 119], we have created a new synthetic dataset to evaluate active learning for object detection in road scenes. In particular, we modified the SYNTHIA environment [108] to generate the SYNTHIA-AL dataset[‖] using Unity Pro game engine. The aim is having an unbalanced foreground/background distribution, simulating the real collection scenario of a driving car. Moreover, a set of object classes and conditions should be predominantly present, while other classes and conditions must appear less frequent. The data is generated by driving a car in a virtual world consisting of three different areas, namely town, city, and highway. These areas are populated with a variety of pedestrians, cars, cyclists, and wheelchairs, except for the highway which is limited to cars. These dynamic objects are arbitrarily spawned at predefined positions with a given probability and follow randomly predefined paths without leaving each area. Several environmental conditions can be set: season (winter, fall, spring), day time (day or night), and weather (clear or rainy). By default, we always use spring and clear during the day, and only change one condition at a time. Objects with no lights can be hard to visualize during the night, so we only use cars for the night condition. Figure 2.3 shows examples of images in the dataset.

Table 2.1 provides the specification of the dataset. The video sequences are captured at 25 fps with a random length between 10 and 30 seconds. We have generated one subset with the default parameters and three smaller subsets with altered conditions. The first subset consists of 150 sequences, which amounts to 75% of all the data, with the default settings, i.e. containing cars, pedestrians, and cyclists, under different daily conditions, but only in the city and highway areas. The

[‖]Available at http://www.synthia-dataset.net

second subset contains 36 sequences (20% of the dataset) captured in the town area instead. The night condition only represents 3% of the whole data (6 sequences) and it is fully contained in the third subset. Finally, we have added wheelchairs and removed cars in the fourth subset, which represents the 2% of the dataset with only 5 sequences. The test set contains 85 sequences with balanced distributions on areas and conditions (except winter) on the three main classes plus another 12 sequences including wheelchairs. All images are automatically annotated with 2D bounding boxes and class labels for every object that can be reasonably seen (more than 50 pixels). Figure 2.6 and 2.7 show few example frames in SYNTHIA-AL dataset.

2.5 Experimental Setup

2.5.1 Active learning procedure

We use a state-of-the-art object detector based on Faster R-CNN [107]. This two-stage detection model first generates object proposals using a sub-network called Region Proposal Generator (RPN) and then outputs predictions for each proposal. It first processes the image through multiple convolutional layers to generate feature maps. After the RPN generates its candidate proposals, a Region of Interest (RoI) pooling layer extracts features for each of the proposals. The extracted features are further processed to obtain classification scores as well as bounding-box regression values, which refine the proposals to localize the objects more accurately.

We start with the model pre-trained on COCO [87], which contains 80K images from 80 different object categories. The initial labeled set $\mathscr{D}_L$ consists of 2% of train dataset that is selected randomly once for all the methods. At each cycle, we fine-tune the latest model of the previous cycle, as we have experimentally observed that this leads to faster convergence than fine-tuning the initial model or from scratch as in [22]. We have also seen that in order not to get stuck in local minima, the learning rate should be high enough. Once the new model is fine-tuned, we use it with the corresponding acquisition function to select b/C frames, which are then labeled and added to $\mathscr{D}_L$. We continue for C cycles until budget b is completely exhausted. In all experiments, the budget per cycle is 2% of the dataset.

Evaluation. For each cycle, we evaluate the model trained with the updated labeled set for that cycle on the test set. Detections are processed using Non-Maxima Supression [40] and thresholded by score, rejecting all detections below 0.5. We use AP averaged over all classes using a detection threshold of IoU> 0.5.

Implementation details. We used Tensorflow's Object Detection API [61] as the base code to develop our experiments. We trained all models with the momentum optimizer with value 0.9 and the initial learning rates 0.02 and 0.001 for SYNTHIA-AL and ImageNet-VID [111] datasets, respectively. We train for 10 epochs and reduce the learning rate by a factor of 5 once after 5 epochs and again at 7 epochs for SYNTHIA-AL. In the case of ImageNet-VID we reduce the learning rate at epochs 3 and 5, training a total of 6 epochs. For efficiency reasons, we resize all images to fixed height of 300 pixels and preserve the aspect ratio. We use a batch size of 12 for all the experiments. Finally, to obtain more stable results we repeat the experiments 3 times and report the mean and standard deviation in our results.

2.5.2 Baselines

Random. Random sampling selects an arbitrary subset of frames from all unlabeled frames. Given the extreme imbalance inherent to video data due to varying video length, uniform random sampling selects most frames from the longer videos while under-representing shorter videos, which damages the performance. Moreover, video data is redundant due to the high similarity between nearby frames, which makes annotating the surrounding frames of an already annotated frame wasteful. For these reasons, we also consider an improved random sampling procedure that includes temporal representativeness, which prevents selecting the k neighbors in both directions of already labeled frames. In the experiments, we set the k to 3 for ImageNet-VID dataset and 1 for SYNTHIA-AL dataset for all the methods. This criterion naturally increases the diversity of the selected batches at each cycle by limiting the similarity between data samples. We call this baseline *Random+R*.

Uncertainty. We consider three other baselines based on uncertainty measures used in recent active learning approaches for object detection [15, 110]. *Least confidence* [82, 110] considers the score of the most probable class and selects those samples that have the lowest score on it. *Entropy* [29] is an information theory measure that captures the average amount of information contained in the predictive distribution, attaining its maximum value when all classes are equally probable. In both cases, we use the average score of all detections in the image to obtain a single score per image. *Margin sampling* [15, 116] uses the difference between the two classes with the highest scores as a measure of proximity to the decision boundary. Following [15], we sum all margin sampling scores of individual detections to aggregate them into an overall image score.

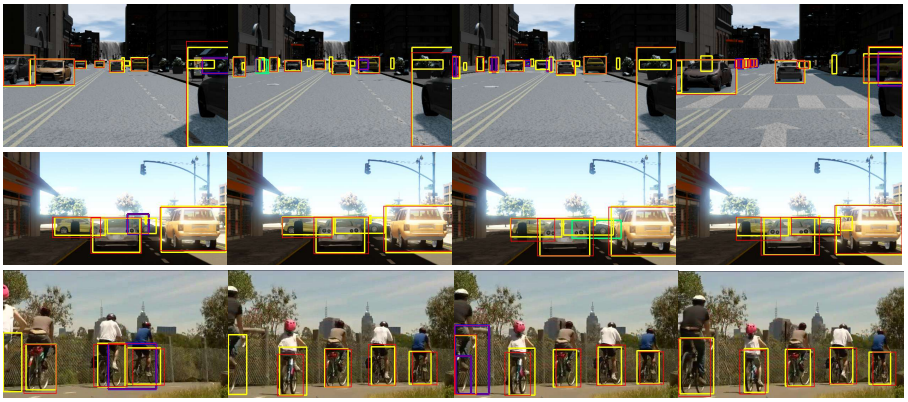


Figure 2.3 – **Examples of errors detected by our temporal coherence approach on SYNTHIA-AL (top, middle) and ImageNet-VID [111] (bottom)**. We show ground-truth boxes in yellow and output detections in red. After solving our graphical model based on temporal coherence, some of the detections are considered as false positives (purple), while other boxes are added as false negatives (green).

2.5.3 Datasets

Besides our SYNTHIA-AL dataset (sec. 2.4), we also perform experiments on a real-image dataset, ImageNet-VID [111], which is commonly used as video object detection benchmark. Since the focus of this chapter is video object detection in road scenes, we select 3 classes that are likely to be encountered in the context of autonomous driving, namely: car, bike, and motorcycle. Selecting all videos that contain these classes amounts to 795 videos in the training set and 87 videos in the validation set, which we use for test. The length of the videos varies between a few frames to over 1000. We have cleaned this dataset by manually discarding all those frames that had missing annotations, which amounts to 20K frames in the training set and 5K frames in the validation set. The final dataset contains 129K frames for training and 14K frames for validation.

2.6 Results

We present active learning results using performance (mAP) curves as a function of the number of selected samples, as usually reported in the literature [44, 115]. This allows us assess the benefit of each active learning method for different total number of samples used to train the model. For each method, we plot the average

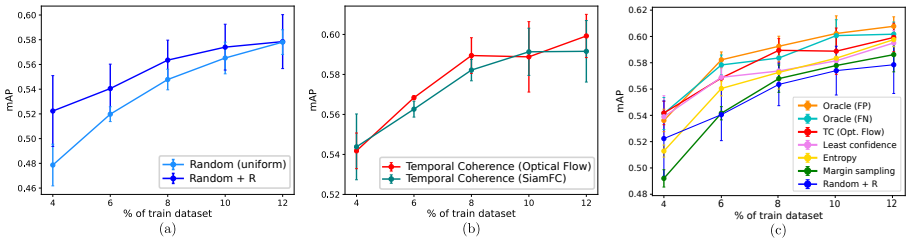


Figure 2.4 – **Results on SYNTHIA-AL.** (a) Random baselines with and without representativeness. (b) Our Temporal Coherence using either Optical Flow or SiamFC. (c) Baselines, oracle-based acquisition, and Temporal Coherence. All curves are the average of 3 runs.

performance for all runs with vertical bars to represent the standard deviation.

We first validate the ability of our graphical model (sec. 2.3.2) to estimate detection errors using temporal coherence. Fig. 2.3 presents some resulting predictions on both datasets. We can see how many FP (purple) are correctly detected, including those corresponding to double detections (top row, rightmost column). Moreover, FN (green) are discovered due to the forward and backward tracking of surrounding detections (middle row, third column).

2.6.1 SYNTHIA-AL

Fig. 2.4 presents all quantitative results on our SYNTHIA-AL dataset. We start by evaluating the difference between the two random baselines: uniform and our enhanced Random+R baseline (Fig. 2.4a). We can observe how the addition of representativeness is clearly beneficial for active learning in video object detection. In the remainder of the chapter, we always include temporal representativeness and per-video sampling for all evaluated methods.

Next, we evaluate the effect of the two types of trackers considered in our temporal coherence method, SiamFC [11] and Optical Flow [120], within the active learning cycles. Fig. 2.4b presents the quantitative evaluation of temporal coherence with either tracker. The results show that there is no improvement gained by using the more sophisticated SiamFC tracker compared to Optical Flow. Furthermore, Optical Flow can significantly speed up the active learning process. In this case, the motion vectors are computed once at the beginning of the process, whereas SiamFC needs to perform expensive computations at every cycle. Finally, we compare Temporal Coherence (TC) with all baselines. To explore an upper bound for TC, we also consider the oracle-based methods of section 2.3.1, selecting those frames with the highest number of FP or FN based on ground-truth informa-

Methods	SYNTHIA-AL		ImageNet-VID	
	mAP	Rel.	mAP	Rel.
All data	0.628	100%	0.839	100%
Random+R	0.578	92.0%	0.821	97.8%
Least Confidence	0.595	94.7%	0.818	97.4%
Margin sampling	0.586	93.3%	0.820	97.7%
Entropy	0.597	95.0%	0.821	97.8%
Oracle (FP)	0.607	96.6%	-	-
Oracle (FN)	0.601	95.7%	-	-
Temporal Coherence (SiamFC)	0.591	94.1%	-	-
Temporal Coherence (Opt. Flow)	**0.599**	**95.3%**	**0.830**	**98.9%**

Table 2.2 – **Active learning results.** The first row shows the performance (mAP) obtained when using the entirety of the dataset. All other rows show the performance of all methods using 12% of all data in SYNTHIA-AL and 10% of ImageNet-VID [111], both in absolute performance and relative to using all data.

tion. These methods are designated by Oracle (FP) and Oracle (FN), respectively. The results in Fig. 2.4c show that our TC method outperforms all three uncertainty based methods and the random baseline. The narrow gap between our TC method and the oracle-based methods implies that FP and FN predictions of the graphical model are effective estimates of the actual errors that the model can learn from. Moreover, TC enables us to achieve more than 95% of performance of the model trained on entire dataset by annotating only 12% of the data. Table 2.2 shows the effectiveness of active learning methods in videos by using a small portion of datasets.

2.6.2 ImageNet-VID

To evaluate our temporal coherence method on a dataset of real images, we perform experiments on ImageNet-VID [111]. Fig. 2.5 compares TC with Optical Flow against uncertainty and random baselines. The results illustrate that TC is superior to all the baselines for all cycles. Additionally, Table 2.2 shows that TC manages to attain almost the full performance of a model trained with the entire dataset by using only 10% of the data, which is a significant reduction in the annotation effort.

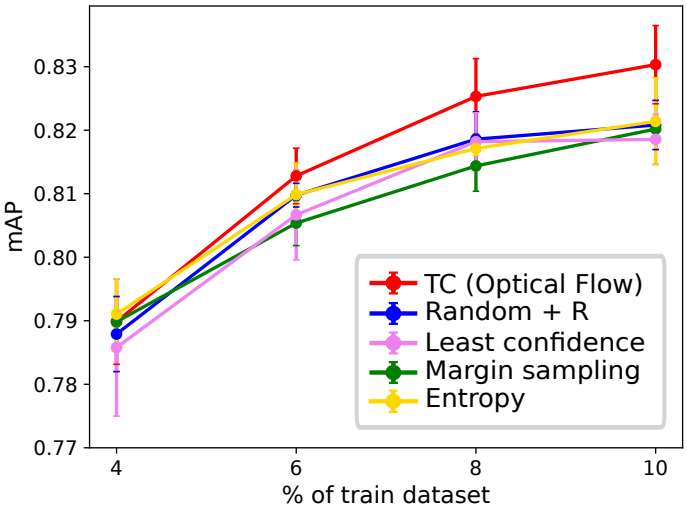


Figure 2.5 – **Results on ImageNet-VID [111].** Average of 3 runs.

2.7 Conclusions

In this chapter, we introduced a novel active learning approach for object detection in videos which exploits the temporal coherence. Our approach is formulated in terms of an energy minimization function of a graphical model built on tracked object detections. Additionally, we introduced a new synthetic dataset specially designed to evaluate active learning for object detection in the context of autonomous driving. Experimental results conducted on two datasets showed that our approach outperformed major active learning baselines. A drawback of temporal coherence based active learning is that it is computationally more demanding than the baselines. We plan to minimize the computational overhead of our system in future research.

Figure 2.6 – Examples of day, night, rainy weather conditions together with city, town frames containing cyclists and cars.

Figure 2.7 – Examples of sunny and rainy weather conditions together with city, town frames containing pedestrians, wheelchair and cars.

3 When Deep Learners Change Their Mind: Learning Dynamics for Active Learning*

Summary: *Active learning aims to select samples to be annotated that yield the largest performance improvement for the learning algorithm. Many methods approach this problem by measuring the informativeness of samples and do this based on the certainty of the network predictions for samples. However, it is well-known that neural networks are overly confident about their prediction and are therefore untrustworthy source to assess sample informativeness. In this chapter, we propose a new informativeness-based active learning method. Our measure is derived from the learning dynamics of a neural network. More precisely we track the label assignment of the unlabeled data pool during the training of the algorithm. We propose to capture the learning dynamics with a metric called label-dispersion, which is low when the network consistently assigns the same label to the sample during the training of the network and high when the assigned label changes frequently. We show that label-dispersion is a promising predictor of the uncertainty of the network, and show on two benchmark datasets that an active learning algorithm based on label-dispersion obtains excellent results. Moreover we employ label-dispersion in the semi-supervised scenario and conduct experiments to evaluate its effeciveness on CIFAR10 and CIFAR100 datasets.*

3.1 Introduction

Deep learning methods obtain excellent results for many tasks where large annotated dataset are available [77]. However, collecting annotations is both time and labor expensive. Active Learning(AL) methods [117] aim to tackle this problem by reducing the required annotation effort. The key idea behind active learning is that a machine learning model can achieve a satisfactory performance with a subset of the training samples if it is allowed to choose which samples to label. In AL, the model is trained on a small initial set of labeled data called initial label pool. An

*This chapter is based on a publication in International Conference on Computer Analysis of Images and Patterns (CAIP), 2021 [8]

acquisition function selects the samples to be annotated by an external oracle. The newly labeled samples are added to the labeled pool and the model is retrained on the updated training set. This process is repeated until the labeling budget is exhausted.

One of the main groups of approaches for active learning use the network uncertainty, as contained in its prediction, to select data for labelling [23, 117, 128]. However, it is known that neural networks are overly confident about their predictions; making wrong predictions with high certainty [99]. In this chapter, we present a new approach to active learning. Our method is based on recent work of Toneva et al. [121], who study the learning dynamics during the training process of a neural network. They track for each training sample the transitions from being classified correctly to incorrectly (or vice-versa) over the course of learning. Based on these learning dynamics, they characterize a sample of being 'forgettable' (if its class label changes from subsequent presentation) or 'unforgettable' (if the class label assigned is consistent during subsequent presentations). Their method is only applicable for labeled data (and therefore not applicable to active learning) and was applied to show that redundant (forgettable) training data could be removed without hurting network performance.

Inspired by this work, we propose a new uncertainty-based active learning method which is based on the learning dynamics of a neural network. With learning dynamics, we refer to the variations in the predictions of the neural network during training. Specifically, we keep track of the model predictions on every unlabeled sample during the various epochs of training. Based on the variations of the predicted label of samples, we propose a new active learning metric called *label-dispersion*. This way, we can indirectly estimate the uncertainty of the model based on the unlabeled samples. We will directly use this metric as the acquisition function to select the samples to be labeled in the active learning cycles. Other than the forgetfulness measure proposed in [121], we do not require any label information.

Experimental results show that label-dispersion better resemble the true uncertainty of the neural networks, i.e. samples with low dispersion were found to have a correct label prediction, whereas those with high dispersion often had a wrong prediction. Furthermore, in experiments on two standard datasets (CIFAR 10 and CIFAR 100) we show that our method outperforms the state-of-the-art methods in active learning.

3.2 Related work

The most important aspect for an active learner is the strategy used to query the next sample to be annotated. These strategies have been successfully applied to

a series of traditional computer vision tasks, such as image classification [43, 45], object detection [2, 7], image retrieval [147], remote sensing [31], and regression [69].

Pool based methods are grouped into three main query strategies relying mostly on heuristics: informativeness [17, 44, 139], representativeness [115], and hybrid [63, 138], and performance-based [43, 49, 137]. A comprehensive survey of these frameworks and a detailed discussion can be found in [117]. **Informativeness-based methods:** Among all the aforementioned strategies, the informativeness-based approaches are the most successful ones, with uncertainty being the most used selection criteria in both bayesian [44] and non-bayesian frameworks [139]. In [83, 142], the authors employed a loss module to learn the loss of a target model and select the images based on their output loss. More recently, query-synthesizing approaches have used generative models to generate informative samples [94, 95, 151]. **Representativeness-based methods:** In [118] the authors rely on selecting few examples by increasing diversity in a given batch. The Core-set technique was shown to be an effective representation learning method for large scale image classification tasks [115] and was theoretically proven to work best when the number of classes is small. However, as the number of classes grows, its performance deteriorates. Moreover, for high-dimensional data, using distance-based representation methods, like Core-set, appears to be ineffective because in high-dimensions p-norms suffer from the curse of dimensionality which is referred to as the distance concentration phenomenon in the computational learning literature [35]. **Hybrid methods:** Methods that aim to combine uncertainty and representativeness use a two-step process to select the points with high uncertainty as of the most representative points in a batch [85]. A weakly supervised learning strategy was introduced in [128] that trains the model with pseudo labels obtained for instances with high confidence in predictions. While most of the hybrid approaches are based on a two-step process, in [130] they propose a method to select the samples in a single step, based on a generative adversarial framework. An image selector acts as an acquisition function to find a subset of representative samples which also have high uncertainty.

3.3 Active learning for image classification

We describe here the general process of active learning for the image classification task. Given a large pool of unlabeled data U and an annotation budget B, the goal of active learning is to select a subset of B samples to be annotated as to maximize the performance of an image classification model. Active learning methods generally proceed sequentially by splitting the budget in several cycles. Here we consider the

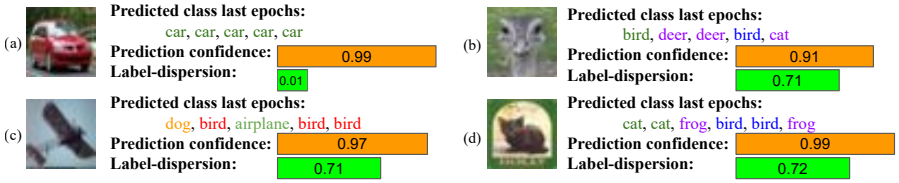


Figure 3.1 – **Comparison between the dispersion and confidence scores.** We show four examples images together with the predicted label for the last five epochs of training. The last predicted label is the network prediction when training is finished. We also report the prediction confidence and our label-dispersion measure. (a) Shows an example which is consistently and correctly classified as *car*. The confidence of model is 0.99 and the consistent predictions every epoch result in low dispersion score of 0.01. (b-d) present examples on which the model is highly confident despite a wrong final prediction and constant changes of predictions across the last epochs. This network uncertainty is much better reflected by the high label-dispersion scores.

batch-mode variant [115], which annotates multiple samples per cycle, since this is the only feasible option for CNN training. At the beginning of each cycle, the model is trained on the initial labeled set of samples. After training, the model is used to select a new set of samples to be annotated at the end of the cycle via an acquisition function. The selected samples are added to the labeled set $\mathscr{D}_L$ for the next cycle and the process is repeated until the total annotation budget is spent.

3.3.1 Label-dispersion acquisition function

In this section, we present a new acquisition function for active learning. The acquisition function is the most crucial component and the main difference between active learning methods in the literature. In general, an acquisition function receives a sample and outputs a score indicating how valuable the sample is for training the current model. Most of informativeness-based active learning approaches consider to assess the certainty of the network on the unlabeled data pool which is obtained after training on the labeled data [23, 117, 128].

In contrast, we propose to track the labels of the unlabeled samples during the course of training. We hypothesize that if the network frequently changes the assigned label, it is unsure about the sample, therefore the sample is an appropriate candidate to be labeled. In figure 3.1 we depict the main idea behind our method and compare it to network confidence. While the confidence score is used to assign the label based on the certainty of the last epoch, the dispersion uses the prediction over all epochs in order to assess the certainty. The first example shows

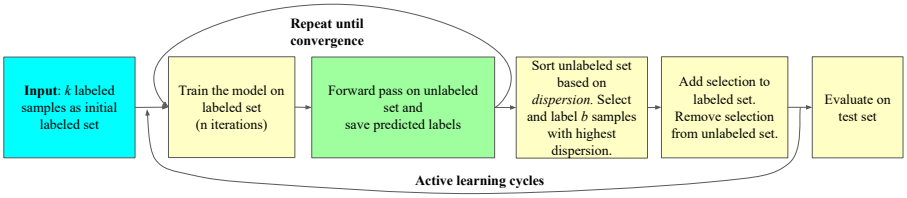


Figure 3.2 – **Active learning framework using Dispersion.** Active learning cycles start with initial labeled pool. The model trained on labeled pool is used to output the predictions and compute dispersion for each sample. The samples with highest dispersion are queried for labeling and added to labeled set. This cycle repeats until the annotation budget is exhausted.

the case of a correct label prediction when both confidence score and dispersion agree. However, in the other three examples, we depict situations where the system predicts the wrong label with high certainty. However, a large dispersion value (i.e. high uncertainty) is the indication of an erroneous prediction.

This idea is based on the concept of *forgettable samples* recently introduced by [121]. [121] states that there exist a large number of unforgettable samples that are never forgotten once learnt. It is shown that they can be omitted from the training set while the generalization performance is maintained. Therefore it suffices to learn the forgettable samples in the train set. However to identify forgettable samples the ground-truth labels is needed. Since we do not have access to the labels in active learning, we propose to use a measure called the *label-dispersion*. The dispersion of a nominal variable is calculated as the proportion of predicted class labels that are not the modal class prediction [41]. It estimates the uncertainty of the model by measuring the changes in the predicted class as following:

$$Dispersion(x) := 1 - \frac{f_x}{T}, \tag{3.1}$$

with

$$\begin{aligned} f_x &= \sum_t 1[y^t = c^*], \\ c^* &= \underset{c=1,\ldots,C}{\arg\max} \sum_t 1[y^t = c], \end{aligned} \tag{3.2}$$

where f_x is the number of predictions falling into the modal class for sample x and C is the number of classes. Larger values for dispersion means more uncertainty in model outputs. Similar to forgettable samples, we are interested in samples for which the model doesn't persistently output the same class.

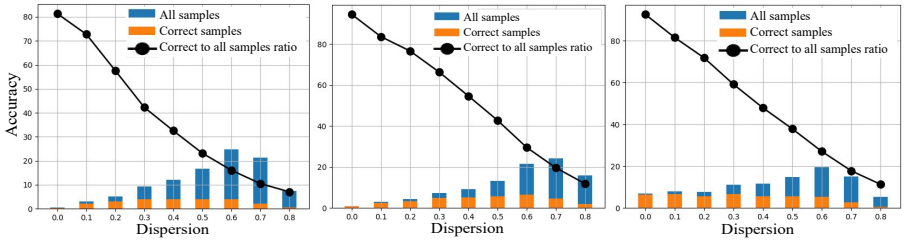


Figure 3.3 – **Informativeness analysis of dispersion across three cycles on CIFAR 100.** The bars show the frequency of samples w.r.t their dispersion. The curve is the ratio between correctly classified and all samples which denote the model is less accurate about the samples with higher dispersion. Hence labeling high dispersion samples provides information that the model lacks.

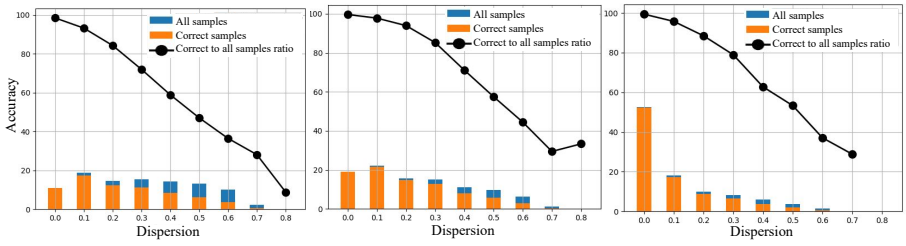


Figure 3.4 – **Informativeness analysis of dispersion across three cycles on CIFAR 10.** The bars show the frequency of samples w.r.t their dispersion. The curve is the ratio between correctly classified and all samples which denote the model is less accurate about the samples with higher dispersion. Hence labeling high dispersion samples provides information that the model lacks.

Fig. 3.2 presents the active learning framework with our acquisition function. During the training of a network at regular intervals we will save the label predictions for all samples in the unlabeled pool (green block in Fig. 3.2). In practice, we will perform this operation at every epoch. These saved label predictions allow us to compute the label-dispersion with Eq. 3.1. We then select the samples with highest dispersion to be annotated and continue to the next active learning cycle until the total label budget is used.

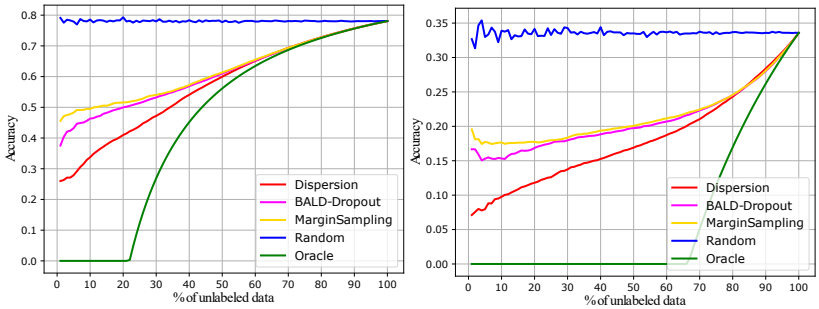


Figure 3.5 – **Informativeness analysis of AL methods on CIFAR10(a) and CIFAR100(b) datasets.** The model is used to infer the label of samples selected by AL methods before labeling and the accuracy is measured. For any amount of unlabeled samples, dispersion offers samples with lower accuracy and hence more informative for the model.

3.3.2 Informativeness Analysis

To assess the informativeness of methods, we compute the scores assigned to the unlabeled samples and sort the samples accordingly. Then we select several portions of the most informative samples (according to their score) and run the model to infer their labels. We argue that annotating the correctly classified samples would not provide much information for the model because the model already knows their label. In contrast, the model can learn from misclassified samples if labeled. We use the accuracy to implicitly measure the informativeness of unlabeled samples. The lower the accuracy, the more informative the samples will be if labeled. Fig. 3.5 shows the accuracy of model on the unlabeled samples queried by each method. The model used in this analysis is trained on the initial labeled set. The accuracy of samples selected randomly remains almost constant regardless of the amount of unlabeled samples. In this analysis, the oracle method by definition uses groundtruth and queries samples that the model misclassified and therefore the accuracy of the model is zero. Among the active learning methods, on both CIFAR10 and CIFAR100 datasets, and for any amount of unlabeled samples, dispersion queries misclassified samples the most, showing that high dispersion correlates well with network uncertainty. These samples can potentially increase the performance of the model if labeled.

To further analyze the dispersion in AL cycles we studied the accuracy of model on the samples selected by dispersion across three cycles. Fig 3.3 illustrates the histogram of samples according to their dispersion score. As can be seen, the samples are mostly populated in mid range dispersion (blue bars). However the ratio

of samples (black curve), that model is accurate about, to all samples is inversely proportional to their dispersion score: the higher dispersion samples we select, the less accurate the model becomes.

3.4 Experimental Results

3.4.1 Experimental Setup

We start with model trained on initial labeled set from scratch and employ Resnet-18 as the model architecture. The initial labeled set consists of 10% of train dataset that is selected randomly once for all the methods. At each cycle, we use the model with the corresponding acquisition function to select b samples, which are then labeled and added to $\mathscr{D}_L$. We continue for 4 cycles until the total budget is completely exhausted. In all experiments, the budget per cycle is 5% and total budget is 30% of the entire dataset. Eventually for each cycle, we evaluate the model on the test set. To evaluate our method, we use CIFAR10 and CIFAR100 [77] datasets with 50K images for training and 10K for test. CIFAR10 and CIFAR100 have 10 and 100 object categories respectively and image size of 32×32. During training, we apply a standard augmentation scheme including random crop from zero-padded images, random horizontal flip, and image normalization using the channel mean and standard deviation estimated over the training set.

Dispersion is computed from the most probable class in the output of the model. During training we do an inference on the unlabeled pool at every epoch and save the model predictions. Based on these predictions we compute the label-dispersion for each sample specifically.

Implementation details. Our method is implemented in PyTorch [100]. We trained all models with the momentum optimizer with value 0.9 and the initial learning rates 0.02. We train for 100 epochs and reduce the learning rate by a factor of 5 once after 60 epochs and again at 80 epochs. Finally, to obtain more stable results we repeat the experiments 3 times and report the mean and standard deviation in our results.

Baselines. We compare our method with several informative and representative-based approaches. *Random sampling:* selects an arbitrary subset of samples from all unlabeled samples. *BALD* [44]: method chooses samples that are expected to maximise the information gained about the model parameters. In particular, it select samples that maximise the mutual information between predictions and model posterior via dropout technique. *Margin sampling* [15]: uses the difference between the two classes with the highest scores as a measure of proximity to the decision boundary. *KCenterGreedy* [115]: is a greedy approximation of KCenter

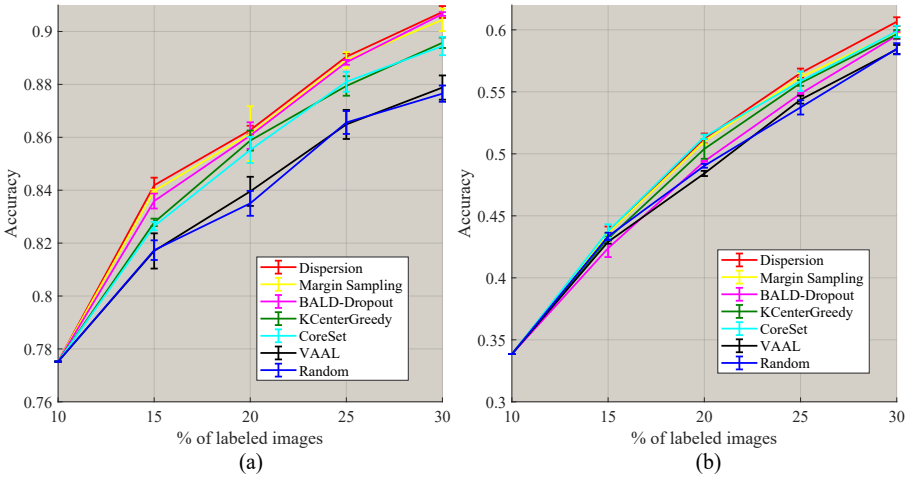


Figure 3.6 – **Performance Evaluation.** Results for several active learning methods on CIFAR10 (a) and CIFAR100 (b) datasets. All curves are average of 3 runs.

problem also known as min-max facility location problem [134]. Samples having maximum distance from their nearest labeled samples in the embedding space are queried for labeling. *CoreSet* [115]: finds samples as the 2-Opt greedy solution of Kcenter problem in the form of Mixed Integer Programming (MIP) problem. *VAAL* [118]: learns a latent space using a Variational Autoencoder (VAE) and an adversarial network trained to discriminate between unlabeled and labeled data. The unlabeled samples which the discriminator classifies with lowest certainty as belonging to the labeled pool are considered to be the most representative and queried for labeling. *Oracle method:* An acquisition function using ground-truth and select samples that the model miss-classified. In order to study the potential of active learning, we evaluate oracle-based acquisition function. Note this is not a useful active learning function in practice, as we would not have access to the ground-truth annotations in a real scenario. In order to make a fair comparison with the baselines, we used their official code and adapted them into our code to ensure an identical setting.

3.4.2 Results

Results on CIFAR10: A comparison with several active learning methods, including both informativeness and representativeness, is provided in Fig. 3.6. As can

be seen in Fig. 3.6(a) dispersion outperforms the other methods across all the cycles on CIFAR10, only the BALD-Dropout method obtains similar results at 30%. The active learning gain of dispersion against Random sampling is around 7.5% at cycle 4, equivalent to annotating 4000 samples less. The informative methods such as Margin Sampling and BALD lie above the representative methods including KCenterGreedy, CoreSet, VAAL and Random highlighting the importance of informativeness on CIFAR10 where the number of classes is limited and each class is well-represented by many samples.

Results on CIFAR100: Fig. 3.6(b) shows the performance of active learning methods on CIFAR100. As can be seen, the methods are closer and the overall performance of Dispersion, Margin sampling and CoreSet are comparable. However, the addition of labeled samples at cycle 3 and 4 makes the dispersion superior in performance to others. The smaller gap between the informative based methods and Random emphasizes the importance of representativeness on CIFAR100 dataset which has more diverse classes that are underrepresented with few samples in small budget size.

Additionally, Table 3.1 illustrates the full performance of models that are trained on the entire datasets. It can be seen, Dispersion manages to attain almost 97% and 82% of full performance on CIFAR10 and CIFAR100 respectively by using only 30% of the data, which is a significant reduction in the labeling effort.

Methods	CIFAR 10		CIFAR 100	
	Acc.	Rel.	Acc.	Rel.
All data	93.61	100%	74.61	100%
Dispersion	**90.74**	96.93%	**60.66**	81.97%
Margin sampling [15]	90.44	96.61%	59.78	80.78%
BALD [44]	90.66	96.85%	59.54	80.46%
KCenterGreedy [115]	89.57	95.69%	59.64	80.59%
CoreSet [115]	89.45	95.56%	59.87	80.91%
VAAL [118]	87.88	93.88%	58.42	78.95%
Random sampling	87.65	93.63%	58.47	79.02%

Table 3.1 – **Active learning results.** Performance of AL methods using 30% of dataset both in absolute performance and relative to using all data.

3.5 Label-Dispersion for Semi-Supervised active learning

As previously discussed in many real applications of large-scale image classification, due to the tedious manual labeling process the labeled data is not enough. Thus developing a framework that combines CNNs and active learning which can jointly learn features and models from unlabeled training data with minimal human annotations has great significance. Usually, incorporating CNNs into active learning framework is not straightforward for real image classification tasks. The labeled training samples given by current AL approaches are often insufficient for CNNs, as the majority unlabeled samples are usually ignored. Active learning, in the small budget schemes, usually selects only few most informative samples (e.g., samples with least confidence) in each cycle and query for the labeling. Thus it is difficult to obtain proper feature representations by fine-tuning CNNs with these minority informative samples. Inspired by the insight from previous works [128], [66] as well as the recently proposed techniques, i.e., self-paced learning [150] we address above mentioned issue by combining the CNN and AL via a complementary sample selection.

We use label-dispersion for sampling pseudo labels in the two-model framework(see Fig. 3.7). The framework has multiple stages: first, we train model A on initial labeled images. Then we use the model A to generate pseudo labels for the unlabeled images. Next, we train the auxiliary model on the combination of oracle labeled images and a portion of pseudo labeled images that are most certain based on label-dispersion or entropy measures. Finally, we run the auxiliary model on the remaining images to select and query the samples for oracle labeling. We iterate this algorithm for a few cycles until the labeling budget is exhausted.

The idea behind using two models is that since at every cycle the pseudo labels might be noisy we treat the model A as a teacher to correct the probably misguided auxiliary model which is trained on both pseudo and oracle labeled samples. Moreover, we replace the pseudo labeled samples in the unlabeled pool after training so they will be available for being picked and assigned more accurate labels in the next cycles.

We use dispersion as the uncertainty measure for pseudo labeling. While for choosing samples to query to the oracle, we used entropy. As explained earlier in this chapter, to compute label-dispersion we do an inference on the unlabeled pool at every epoch during the course of training and save the model predictions. Based on these predictions we compute the label-dispersion for each sample. For the initialization of models we start the framework with training the model A from scratch on the initial labeled pool. After that, we initialize each model with the latest

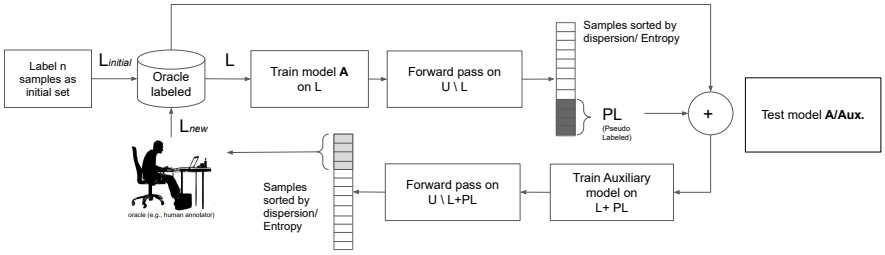


Figure 3.7 – **Our active learning framework exploiting pseudo labels.**

previously trained model.

During training, we apply a standard augmentation scheme including random crop from zero-padded images, random horizontal flip, and image normalization using the channel mean and standard deviation estimated over the training set.

Implementation details. Our method is implemented in PyTorch [100]. We trained all models with the momentum optimizer with value 0.9 and the initial learning rates 0.02. We train for 100 epochs and reduce the learning rate by a factor of 5 once after 60 epochs and again at 80 epochs. Finally, to obtain more stable results we repeat the experiments 3 times and report the mean and standard deviation in our results.

We employ Resnet-18 as the model architecture. The initial labeled set consists of 10% of train dataset that is selected randomly once for all the methods. At each cycle, we use the model with the corresponding acquisition function to select b samples, which are then labeled and added to $\mathscr{D}_L$. We continue for 4 cycles until the total budget is completely exhausted. In all experiments, the budget per cycle is 5% and total budget is 35% of the entire dataset. We incorporate 10% to 25% of the dataset as pseudo labeled samples every cycle. Eventually for each cycle, we evaluate the both models on the test set.

3.5.1 Results

We evaluate the performance of our framework using dispersion and/or entropy against CEAL method on CIFAR10 and CIFAR100 datasets. The performance evaluation of our method on CIFAR10 is shown in Fig. 3.8.

As can be seen the auxiliary models lie above the performance of model A. This is due to the fact that auxiliary model (B) is trained on both pseudo and oracle labeled samples whereas model A is only trained on oracle labeled samples. Besides, our method (shown as B) outperforms the CEAL method either by using 10% or 25%

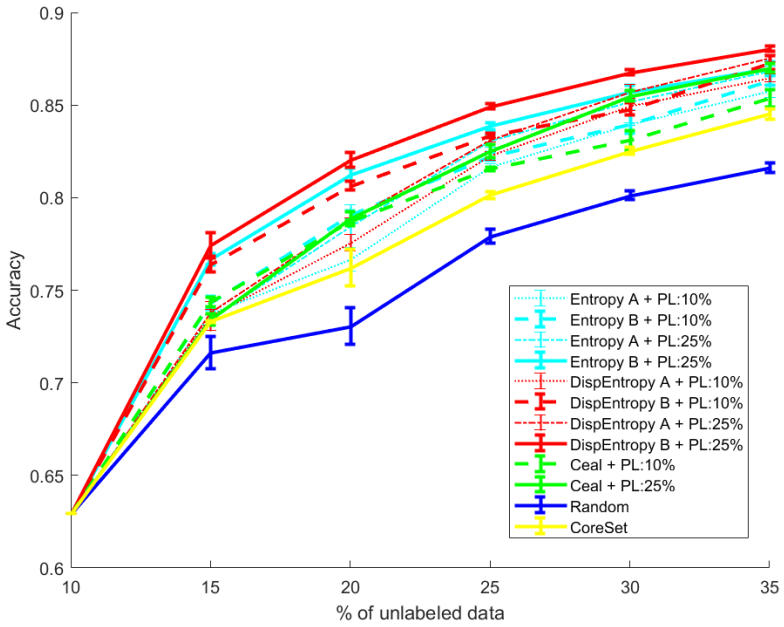


Figure 3.8 – **CIFAR10: Comparison of our method with CEAL and Core-set methods..**

of pseudo labeled samples. Moreover, the best performance is achieved when the dispersion criterion is used to select pseudo labeled samples and entropy is used to select samples for oracle labeling.

The performance evaluation of our method on CIFAR100 is shown in Fig. 3.9. As can be seen the auxiliary models lie above the performance of model A. Again, this is because auxiliary model (B) is trained on both pseudo and oracle labeled samples whereas model A is only trained on oracle labeled samples. In addition, our method (shown as B) outperforms the CEAL method either by using 10% or 25% of pseudo labeled samples. The best performance is achieved when entropy criterion is used for selecting both pseudo and oracle labeled samples.

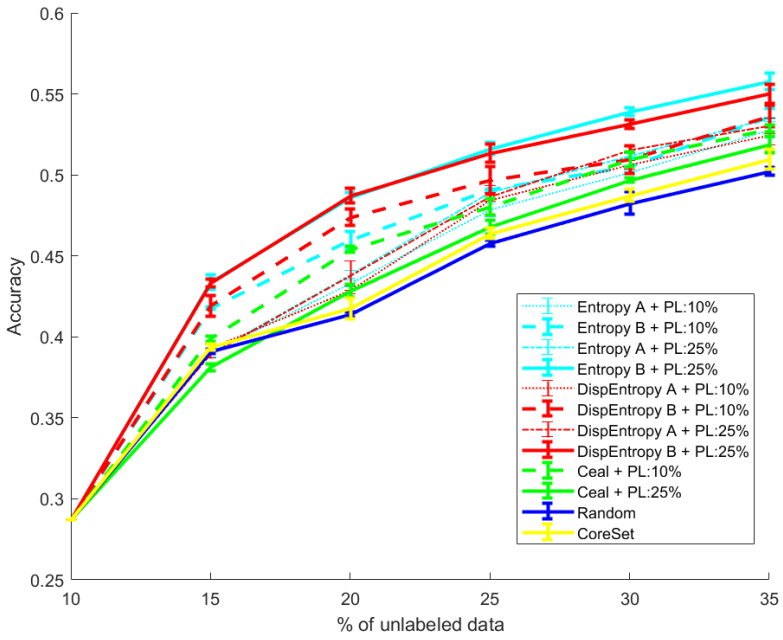


Figure 3.9 – **CIFAR100: Comparison of our method with CEAL and Core-set methods..**

3.6 Conclusion

We proposed an active learning algorithm based on the learning dynamics of neural networks. We introduced the label-dispersion metric, which measures label-consistency during the training process. We showed that this measure obtains excellent results when used for active learning. For future work, we are interested in exploring label-dispersion for other research fields such as out-of-distribution detection and within the context of lifelong learning.

4 Class-Balanced Active Learning for Image Classification *

Summary: *Active learning aims to reduce the labeling effort that is required to train algorithms by learning an acquisition function selecting the most relevant data for which a label should be requested from a large unlabeled data pool. Active learning is generally studied on balanced datasets where an equal amount of images per class is available. However, real-world datasets suffer from severe imbalanced classes, the so called long-tail distribution. We argue that this further complicates the active learning process, since the imbalanced data pool can result in suboptimal classifiers. To address this problem in the context of active learning, we proposed a general optimization framework that explicitly takes class-balancing into account. Results on three datasets showed that the method is general (it can be combined with most existing active learning algorithms) and can be effectively applied to boost the performance of both informative and representative-based active learning methods. In addition, we showed that also on balanced datasets our method generally results in a performance gain.*

4.1 Introduction

Convolutional neural networks have obtained state-of-the-art results on several computer vision tasks such as large-scale object detection [104] or VQA [131]. However, the training of these often very large networks requires large-scale labeled datasets, that are labor intensive and expensive to construct. Generally, in real-world the amount of data that could be labeled is literary unlimited (e.g. in autonomous driving, or robotics applications). Given an initial labeled dataset, deciding what new data to label from the unlabeled data pool is a relevant question. Active learning addresses this research questions and aims to minimize the labeling effort while maximizing the obtained performance of the machine learning algorithm. Active learning has successfully been shown to reduce the labeling effort for image classification [6, 115], object detection [155], regression [69], and semantic

*This chapter is based on a publication in IEEE Winter Conference on Applications of Computer Vision (WACV), 2022 [9]

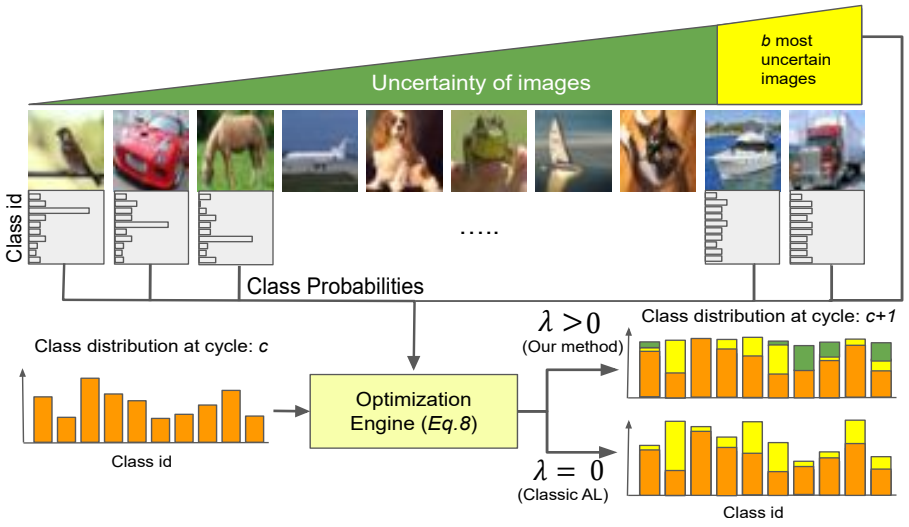


Figure 4.1 – **Overview of our active learning framework.** The unlabeled samples are sorted by their uncertainty from green to yellow in ascending order. Given the the uncertainty of unlabeled samples and class distribution at cycle c, we propose to solve an optimization problem ($\lambda > 0$) yielding samples that are simultaneously informative and form a balanced class distribution for training. Our sampling selects samples with lower uncertainty (in green) in addition to high uncertainty to improve class-balanced profile. In contrast, classical AL methods ($\lambda = 0$) selects the most uncertain samples (in yellow) that result in an informative yet imbalanced training set.

segmentation [47, 127].

Several query strategies have been proposed for sample selection. The most popular ones are those based on informativeness [142] and representativeness [115] which demonstrated to be efficient for the task of selecting the most valuable samples. The informativeness criteria is responsible for selecting those samples which are the most uncertain (usually characterized by high-entropy) because they affect the generalization capability of the model (they are the ones which are mostly confusing the classifier, especially at the start of the active learning process when the number of labeled samples is small), while representativeness guarantees a diversity of the samples, following the underlying data distribution of the unlabeled data pool.

Visual recognition datasets in computer vision research are often almost uniformly distributed (e.g. CIFAR [77] and ILSVRC [79]). However, for many real-world

problems data follows a long-tail distribution [97], meaning that a small number of head-classes are much more common than a large number of tail-classes (e.g. iNaturalist [124], landmarks [98]). Classification on such imbalanced dataset is an important research topic [28, 60, 105]. However, active learning is mostly studied on curated close to uniform datasets. Given the predominance of long-tail distributions, especially for real-world applications in which active learning is a crucial capability, we consider here the study active learning for imbalanced datasets. The aim is to minimize the labeling effort, while maximizing the performance when measured on a balanced test set.

Closely related to the class-imbalance dataset problem, is the sampling bias problem which is a well-documented drawback of active learning [30, 92]. Datasets collected by active learning algorithms break the assumption that the data is identically and independently distributed (i.i.d), since the active learning algorithm might be biased towards particular regions of the unlabeled data manifold. One possible consequence of the sampling bias can be that the distribution over the classes does no longer follow that of the unlabeled data pool. Several papers have investigated this aspect of active learning however it remains not fully understood [12, 39].

To mitigate the problems caused by the sampling bias and imbalanced datasets, in the current chapter we introduce an optimization framework which corrects the class-imbalance presented in the unlabeled data pool, and aims to bias instead our selected samples to resemble the uniform distribution of the test set. The overview of the proposed approach is depicted in figure 4.1. Since we have no access to the class labels of the unlabeled data, we propose to trust the predicted labels, and use them to select a set of class-balanced images. This combination leads to a minimization problem, which can be formalized as a binary programming problem. We show that our optimization scheme is efficient, boosting the performance of both informativeness and representativeness methods. In summary, the main contributions of this chapter are:

- We propose a novel active learning method for imbalanced unlabeled dataset that encourages the selection of class-balanced samples.
- The proposed optimization method is general and can be applied to both informativeness and representativeness based methods.
- Extensive experiments show that our method improves performance of active learning on imbalanced datasets. We show that even for balanced datasets the proposed method can lead to improvements, mostly by countering the sampling bias introduced by active learning.

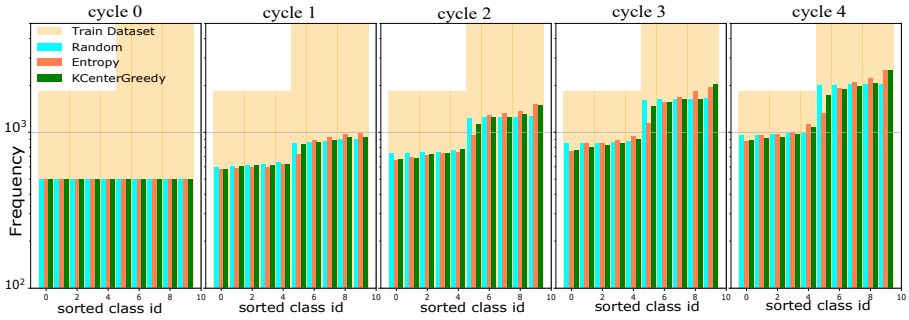


Figure 4.2 – **Biased sampling across four AL cycles.** In the background, the imbalanced dataset is illustrated in yellow. The class distributions for two active learning approaches and random sampling are shown. Similar to Random sampling (in cyan), samples selected by active learning algorithms follow the biased distribution. Results are on imbalanced CIFAR10 (IF=0.3).

4.2 Related Work

Active Learning. Active Learning has been widely studied in various applications such as image classification [43, 45, 72] (including medical image classification [113] and scene classification [87]), image retrieval [147], image captioning [32], object detection [155], and regression [42, 69].

Over the past two decades, several different strategies have been proposed for sample query, which can be divided in three main categories: informativeness [8, 17, 44, 50, 140], representativeness [113, 115] and hybrid approaches [63, 138]. A comprehensive survey of these frameworks and a detailed discussion can be found in [117].

Among all the aforementioned strategies, the informativeness-based approaches are the most successful ones, with uncertainty being the most used selection criteria used in both bayesian [44] and non-bayesian frameworks [85]. In [44], they obtain uncertainty estimates through multiple forward passes with Monte Carlo Dropout, but it is computationally inefficient for recent large-scale learning as it requires dense dropout layers that drastically slow down the convergence speed. More recently, [3] measures the uncertainty of the model by estimating the expected gradient length. On the other hand, [83, 142] employ a loss module to learn the loss of a target model and select the images based on their output loss.

Representativeness-based methods rely on selecting examples by increasing diversity in a given batch [36]. The Core-set technique [115] selects the samples by

minimizing the Euclidian distance between the query data and labeled samples in the feature space. The Core-set technique is shown to be an effective representation learning method, however, its performance is limited by the number of classes in the dataset. Furthermore, Core-set, like other distance-based approaches, are less effective due to feature representation in high-dimensional spaces since p-norms suffer from the curse of dimensionality [35]. In a different direction, [118] uses an adversarial approach for diversity-based sample query, which samples the data points based on the discriminator's output, seen as a selection criteria.

Class-Imbalanced Data. Learning with class-imbalanced data is a well investigated research problem [65]. There are several approaches to address the conflict between a highly imbalanced training dataset and the objective to perform equally well for all classes on the test set. The bias towards the most frequent classes can be reduced by *re-weighting* samples in the training objective. One popular approach is re-weighting samples by the inverse of their class-frequency [60]. Cui et al. [28] improve upon this method, and propose to re-weight samples with the effective number of its class. Another approach is based on *re-sampling* where samples of rare classes are more often rehearsed during training [54]. Ren et al. [105] investigate the training on imbalanced data in combination with label noise. They propose a method based on meta-learning that learns to assign weights to training examples. Our proposed method aims to prevent the dataset imbalance that could arise during the active learning cycles. Other than the here discussed methods, our approach is not presented with an imbalanced dataset, but actively participates in its construction. We show that incorporating class-balance as one of the objectives of active learning is of key importance on imbalanced datasets.

Previous works that addressed class imbalance in AL include [1, 13, 132, 144]. Among them, only [1] is applied to deep learning. Nevertheless, it studies sequential AL as balancing is performed during manual labeling making it practically infeasible for batch mode AL. In the same line [16] lacks automatic model to address the class-imbalance problem and the evaluations are human-centered only. Unlike [24] that lacks evaluation on large scale dataset, we show our method extends to Tiny ImageNet as a large dataset with diverse classes.

4.3 Class Imbalance in Active Learning

4.3.1 Active Learning Setup

Given a large pool of unlabeled data $\mathscr{D}_{\mathscr{U}}$ and a total annotation budget B, the goal is to select b samples in each cycle to be annotated to maximize the performance of a classification model. In general, AL methods proceed sequentially by splitting

the budget in several *cycles*. Here we consider the batch-mode variant [117], which annotates b samples per cycle, since this is the only feasible option for CNN training. At the beginning of each cycle, the model is trained on the labeled set of samples $\mathscr{D}_{\mathscr{L}}$. After training, the model is used to select a new set of samples to be annotated at the end of the cycle via an *acquisition function*. The selected samples are added to the labeled set $\mathscr{D}_L$ for the next cycle and the process is repeated until the annotation budget b is spent. The acquisition function is the most crucial component and the main difference between AL methods in the literature. In the remainder of this section, we describe our contributions to acquisition functions.

4.3.2 Motivation

Most active learning methods propose efficient sampling methods that are class agnostic. The underlying assumption is that the distribution of train and test datasets are uniform. However, in real world scenarios, where the datasets might be heavily imbalanced, the methods suffer from biased sampling towards the majority class. AL methods tend to sample more from frequent classes and less from minority classes which consequently leads to biased predictions and a performance drop. Fig. 4.2 presents an example of a such dataset with various AL methods (see Fig. 4.8,4.9,4.10 for more distributions). As it can be seen, the distribution of samples selected by both informative and representative based methods follow the distribution of the unlabeled dataset. Moreover the imbalance of selected samples grows across the cycles. It is known that when we aim for good performance on all classes these imbalanced training sets are suboptimal [28, 60]. We tackle the problem of class imbalance in the remainder of this section.

4.3.3 Reducing Class Imbalance

A balanced set of samples requires an equal number of samples per class. Since we have no access to the class labels, we make an estimate of distribution of samples by using a probability matrix. Assume we have $|\mathscr{D}_{\mathscr{U}}| = N$ unlabeled samples in C categories. We use the classifier to output the softmax probability matrix P on the unlabeled samples:

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1C} \\ p_{21} & p_{22} & \cdots & p_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NC} \end{bmatrix} \in R^{N \times C} \tag{4.1}$$

where each row sums to 1. Similar to [37], we use variable $z_i \in \{0,1\}$ associated to

sample i to indicate whether a sample i is selected or not. To measure the distance between the estimated distribution and the desired distribution we employ $\ell 1$ norm as:

$$\ell 1(\Omega, P^T \boldsymbol{z}) = \|\Omega(c) - P^T \boldsymbol{z}\|_1. \tag{4.2}$$

Here $\Omega(c)$ is vector with components specifying the number of required samples from each class in order to achieve balance at cycle c. Given the labels of samples selected in previous cycles, it is straightforward to compute the samples required at cycle c:

$$\Omega(c) = [\omega_1, \omega_2, ..., \omega_C], \tag{4.3}$$

where,

$$\omega_i = max(\frac{cb + b_0}{C} - n_i, 0), \tag{4.4}$$

b is the budget per cycle, b_0 is the size of the initial labeled set, $c \in \{1, 2, 3, ...\}$ denotes the cycle, and n_i is the number of samples selected from class i in previous cycles. Condition 4.4 avoids oversampling from a particular class. To obtain Ω at cycle $c = 1$ for instance, given that we start the AL cycles from uniform initial set with $n_i = b_0/C$ we have:

$$\Omega(1) = \frac{b}{C} \mathbb{1}_{C \times 1} \tag{4.5}$$

In the following, we will minimize Eq. 4.2 to encourage the selection of class-balanced samples.

4.4 Class Balanced Active Learning

In this section, we introduce the Class Balanced Active Learning (CBAL) formulation for classification.

4.4.1 Informativeness

Entropy We describe our optimization framework that selects the most uncertain samples while seeking to balance the number of samples over classes. Based on informativeness approach, given the probability matrix the goal is to find samples that are most uncertain for the model. To measure the uncertainty we use *Entropy* [29] as an information theory measure that captures the average amount of information contained in the predictive distribution, attaining its maximum value when all classes are equiprobable. Given the softmax probabilities, the entropy of a

sample is computed as:

$$H = -\sum_{i=1}^{C} p_i \log p_i. \tag{4.6}$$

We aim to select samples with maximum entropy. Consequently, the sum of candidates' entropy should also be maximized. In matrix notation form, this is expressed as:

$$\sum_{\{j|z_j=1\}} H(x_j) = -\boldsymbol{z}^T (P \odot \log(P)) \mathbb{1}_{C\times 1}, \tag{4.7}$$

where $\boldsymbol{z}$ is all-ones column vector and $\odot$ denotes element wise multiplication. $\mathbb{1}_{C\times 1}$ is an all-ones column vector. In our objective we will minimize the negative entropy, which is equal to maximizing the entropy.

Finally, we combine the informative and balancing objectives in a single optimization problem given as:

$$\begin{aligned} \min_{z} \quad & \boldsymbol{z}^T (P \odot \log(P)) \mathbb{1}_{C\times 1} + \lambda \|\Omega(c) - P^T \boldsymbol{z}\|_1 \\ \text{s.t.} \quad & \boldsymbol{z}^T \mathbb{1}_{N\times 1} = b, \quad z_i \in \{0,1\}, \quad \forall i = 1,2,...,N \end{aligned} \tag{4.8}$$

where λ is a parameter that regularizes the contribution of the balancing term in the objective. Minimizing the cost in Eq. 4.8 encourages to select sufficient number of samples per class while choosing the most informative ones. The cost function consists of an affine term and a $\ell 1$ norm that are both convex, and subsequently their linear combination is also convex. However, as the constraint is non-convex the optimization problem becomes non-convex. The underlying problem is Binary Programming that can be optimally solved by an off-the-shelf optimizer using LP relaxation and the branch and bound method. Algorithm 10 presents the AL cycles using our approach.

Regularizer λ. Next, we analyze the effect of varying parameter λ on the cost function. We start with a model trained on initial labeled samples of CIFAR100 dataset. Then, for every λ in range $(0,3)$ the cost function in Eq. 4.8 is minimized. Fig. 4.4 illustrates the changes in entropy loss and the $\ell 1$ loss as the components of the cost function with respect to λ. For comparison purposes, the horizontal lines represent the same losses measured on samples given by standard entropy and entropy L1-pseudo label methods. The latter uses the hard labels given by the model to unlabeled samples also known as "Pseudo Labels" for balancing. See Fig. 4.7 for more details and performance evaluation of Entropy-L1-Pseudo Label. As can be seen, greater λ reduces entropy, $\ell 1$, and $L1score$ (introduced in 4.5.1). It is

Algorithm 1 Class Balancing AL

Input: Unlabeled Pool $\mathscr{D}_{\mathscr{U}}$, Total Budget B, Budget Per Cycle b,
Initialize: Initial labeled pool $|\mathscr{D}_{\mathscr{L}}| = b_0, c = 1$

while $|\mathscr{D}_{\mathscr{L}}| < B$ **do**
  Train CNN classifier Θ on $\mathscr{D}_{\mathscr{L}}$
  Use Θ to compute probabilities for $x \in \mathscr{D}_{\mathscr{U}}$
  Compute $\Omega(c)$ from Eq. 4.3
  Solve 4.8 or Algorithm 2 for greedy, to obtain z
  Query z to $\mathscr{ORACLE}$
  $\mathscr{D}_{\mathscr{L}} \leftarrow \mathscr{D}_{\mathscr{L}} \cup z, \quad \mathscr{D}_{\mathscr{U}} \leftarrow \mathscr{D}_{\mathscr{U}} \setminus z$
  $c \leftarrow c + 1$
end while
return $\mathscr{D}_{\mathscr{L}}, \Theta$

notable that the samples selected with greater λ are more balanced but at the cost of lower entropy. As a result, there is a trade-off between balancedness and entropy of samples.

Variational Adversarial Active Learning (VAAL) VAAL [118] is considered to be one of the current state-of-the-art algorithms on active learning. This model uses a variational autoencoder to map the distribution of labeled and unlabeled data to a latent space. A binary adversarial classifier (analogous to a GAN discriminator) is trained to predict if an image belongs to the labeled or the unlabeled pool. The unlabeled images which the discriminator classifies with lowest certainty as belonging to the labeled pool are considered to be the most representative with respect to other samples which the discriminator thinks belong to the labeled pool. Thus, the images labeled by the discriminator with lower certainty are sampled to be labeled in the next cycle. Considering the uncertainty estimate u of the discriminator, we can encourage finding a balanced sample set by minimizing:

$$\begin{aligned} &\min_{z} \quad \boldsymbol{z}^T u + \lambda \|\Omega(c) - P^T \boldsymbol{z}\|_1 \\ &\text{s.t.} \quad \boldsymbol{z}^T \mathbb{1}_{N\times 1} = b, \quad z_i \in \{0,1\}, \quad \forall i = 1,2,...,N \end{aligned} \tag{4.9}$$

Bayesian Active Learning with Disagreement BALD method chooses samples that are expected to maximise the information gained about the model parameters. In particular, it select samples that maximise the mutual information between predictions and model posterior [44]. It approximates Bayesian inference by drawing Monte Carlo sampling via dropout. Similar to our previous approach, we summa-

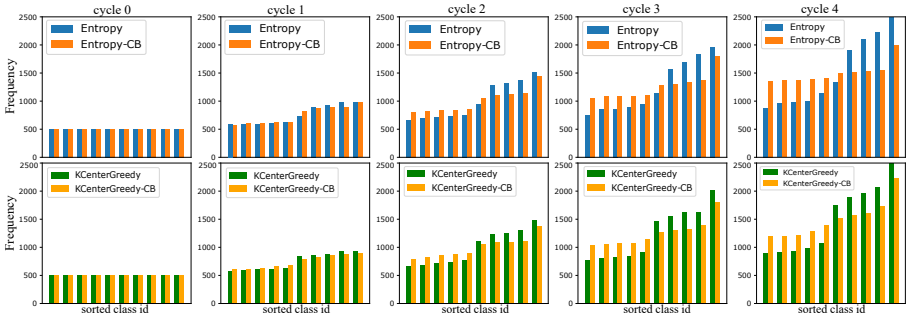


Figure 4.3 – **Class balanced sampling.** Class distribution for Entropy and KCenterGreedy for several active learning cycles on imbalanced CIFAR10 (IF= 0.3.). Our proposed class-balancing (CB) method results in a improved class-balance for both methods.

rize the mutual information assigned to samples into a vector and incorporate into our optimization problem.

4.4.2 Representativeness

Representativeness-based methods aim to increase the diversity of the selected batch [115]. These active learning approaches select the samples iteratively one at a time. In fact, every selected sample influences the next one. Therefore, a method that integrates greedy selection while maintaining the class balance of samples is of great interest. For this reason, we present the greedy class balancing algorithm that incorporates balancing in the sample selection.

We focus on a prominent method of this approach namely KCenterGreedy, which is a greedy approximation of KCenter problem also known as min-max facility location problem [134]. Our aim is to find b samples having maximum distance from their nearest labeled samples while keeping the samples class-balanced. Similar to [115], we compute the embeddings for unlabeled samples via a deep neural network. Specifically, we employ the model for inference on unlabeled samples and consider the penultimate fully connected layer as the visual embedding. Then, we compute the geometrical distances between the representations in the embedding space and construct the distance matrix D. Given N unlabeled and L labeled samples, $d_{ij} \in D_{N\times L}$ is the euclidean distance between the embeddings of unlabeled sample i to labeled sample j. The algorithm 8 presents the KCenterGreedy sample selection combined with class balancing. We propose similar cost function to Eq.4.8 for the greedy sampling. In the algorithm $P^T z$ represents the cost of already selected samples and matrix Q represents the unlabeled samples to choose from.

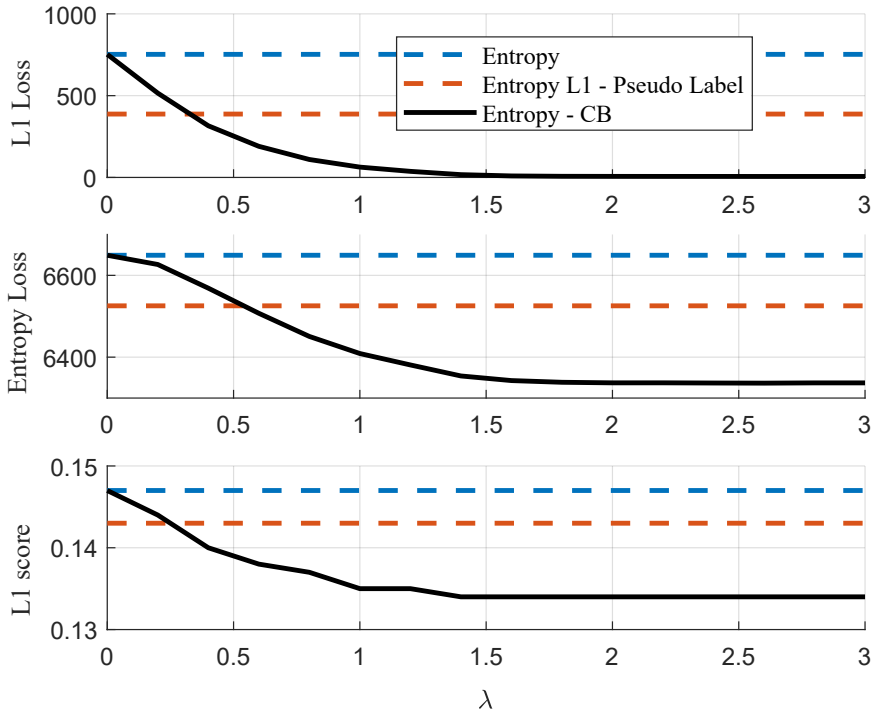


Figure 4.4 – **The effect of λ on L1 and entropy losses in the cost function 4.8.**

The broadcasting within the L1 norm is for the consistency across dimensions of labeled samples, unlabeled samples and thresholds. Although here we integrated the balanced sampling with KcenterGreedy, our method is general and applicable to any greedy acquisition method.

4.5 Experiments

4.5.1 Experimental Setup

We evaluate our method on three image classification benchmarks and the imbalanced variants. The initial labeled set $\mathscr{D}_{\mathscr{L}}$ consists of 10% of the training dataset that is uniformly selected from all classes at random. At each cycle we start with

Algorithm 2 Greedy Class Balancing Selection

Input: Softmax output $P_{N\times C}$, Distance Matrix $D_{N\times L}$, Balancing threshold $\Omega_{C\times 1}$, Regularizer λ, Budget Per Cycle b

Initialize: $z^{(0)} = \mathbf{0}_{N\times 1}, \quad Q = P$

1: **for** $i = 0 : b-1$ **do**

2: $d^{(i)}_{N\times 1} \leftarrow min(D, axis = 1)$ ▷ for each unlabeled sample find the nearest labeled sample

3: $\psi \leftarrow argmin(-d^{(i)}_{(N-i)\times 1} + \lambda \|\Omega(c) - Q^T_{C\times(N-i)} - P^T_{C\times N} z^{(i)} \mathbb{1}_{1\times(N-i)})\|^T_1$

4: $z^{(i+1)}(\psi) \leftarrow 1$ ▷ select the sample

5: $Q \leftarrow P(z^{(i)} = 0, :)$ ▷ keep the remaining unlabeled samples in Q

6: $D \leftarrow D_{(N-i)\times(L+i)}$ ▷ update D by removing a row and adding a column correspond to newly selected sample

7: **end for**

8: **return** $z^{(b)}$

our base model either from scratch or, in case of Tiny-imagenet, we start from a pretrained imagenet model. We train the model in c cycles until the budget B is exhausted. The budget per cycle for all experiments is 5% of the original dataset.

Datasets. To evaluate our method, we use CIFAR10 and CIFAR100 [77] datasets with 50K images for training and 10K for test. CIFAR10 and CIFAR100 have 10 and 100 object categories respectively and an image size of 32×32. To evaluate the scalability of our method we evaluate on Tiny ImageNet dataset [80] with 90K images for training and 10K for testing. There are 200 object categories in Tiny ImageNet with an image size of 64×64.

Long-Tailed Datasets. To verify our approach on imbalanced datasets, we make the CIFAR10, CIFAR100 and Tiny ImageNet class-imbalanced. Again, we reserve 10% of samples of the three datsasets for initial labeled set. As in [27] we create long-tailed datasets with the remaining 90% by randomly removing training examples. In particular, the number of samples drops from y-th class is $n_y \cdot \text{IF}$ where n_y is the original number of training samples in class y and the imbalance factor $\text{IF} \in (0,1)$. For the construction of long-tailed datasets we apply IF to half of the classes, and use $\text{IF} \in \{0.1, 0.3\}$.

Baselines. We compare our method with Random sampling and several informative and representative-based approaches including Entropy sampling, KCenter-

Greedy, VAAL, BALD and Core-set. In order to make a fair comparison with the baselines, we used their official code and adapted them into our code to ensure an identical setting.

Performance Evaluation. We measure the accuracy on the test set to evaluate the performance of the model. Results for all experiments are averaged over 3 runs. For each method we plot the average performance for all runs with vertical bars to represent the standard deviation. To measure the balancedness of selected samples, we use $L1\_score$ by computing $\ell 1$ distance between samples' distribution and uniform distribution. In order to have a measure ranging from 0 to 1, we normalize $\ell 1$ with the factor obtained as following:

$$\begin{aligned} &\ell 1([b,0,,..,0],[\frac{b}{C},...,\frac{b}{C}]) = \\ &|b-\frac{b}{C}|+|0-\frac{b}{C}|+...+|0-\frac{b}{C}| = \frac{2b(C-1)}{C}. \end{aligned} \tag{4.10}$$

The first argument represents the distribution in which the entire budget b is spent to sample from a single class while the second argument represents the uniform sampling.

Implementation details. Our method is implemented in PyTorch[†] [100]. We start with Resnet18 [57] model trained from scratch every cycle. For Tiny-Imagenet dataset however we start with pretrained ImageNet model and Resnet101. All the models are trained with SGD optimizer with momentum 0.9 and an initial learning rate of 0.02 and 0.01 for CIFAR10/100 and Tiny ImageNet respectively. We train CIFAR datasets for 100 epochs and reduce the learning rate by a factor of 0.5 once at 60 and again at 80 epochs. In the case of Tiny ImageNet we reduce the learning rate at 10, 15, 20, 25 epochs by factor of 0.5 training for a total of 30 epochs. During training, we apply a standard augmentation scheme including random crop from zero-padded images, random horizontal flip, and image normalization using the channel mean and standard deviation estimated over the training set. We set the regularizer λ based on the analysis in 4.4 specifically for each method. We choose the smallest λ after which the L1 loss does not diminish further. Once we chose λ we keep it fixed for that method across the experiments. To efficiently solve the optimization problem we used python CVXPY [33] with Gurobi solver [51].

[†]Upon acceptance, we will release the code for our method.

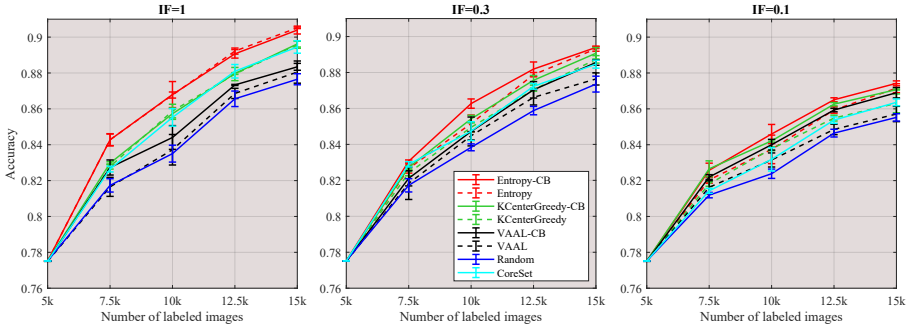


Figure 4.5 – **Performance evaluation.** Results for several active learning methods on CIFAR10 for different imbalance factors (IF).

4.5.2 Experimental Results

Performance on CIFAR10. Fig. 4.3 provides an evaluation of the class balancing technique on Entropy and KcenterGreedy. The distribution of samples selected by Class Balanced (CB) methods evidently remain close to the uniform compared to the baselines across cycles.

Fig. 4.5 presents the quantitative results on CIFAR10. Dashed curves represent the standard methods and solid curves represent the methods equipped with class-balancing. We start by evaluating the performance of the methods on the balanced (original) dataset denoted by IF=1. We observe in Fig. 4.5.a that the addition of class balancing gives similar results compared to the standard methods. However, for the case of VAAL, class-balancing results in notable improvements. Next, we evaluate the performance of class-balancing on the imbalanced CIFAR10 dataset where IF=0.3. Fig. 4.5.b illustrates clearly how class-balancing is beneficial for all the methods across the cycles. As can be seen, the class-balanced variants constantly improve the performance of both informative and representative based baselines. Regarding the active learning gain, Entropy-CB achieves the performance of 86% whereas Random requires almost 10% more annotation equivalent to 5K images to achieve the same performance. In our experiments CoreSet and KCenterGreedy-CB perform similarly on the balanced dataset (IF=1). However, when the dataset is imbalanced (IF=0.3 and IF=0.1) the performance of CoreSet degrades compared to KCenterGreedy-CB. As CoreSet is a MIP (Mixed Integer Programming) problem, our technique cannot be applied to this method.

Fig. 4.5.c illustrates the performance of methods on a severely imbalanced dataset where IF=0.1. We observe a considerable improvement in methods with class balancing over the baselines. In particular VAAL-CB achieves a growing im-

provement of 1% on average over VAAL across the cycles. See table 4.1 for the details of performance gains over baselines.

Imbalance Factor	Methods	Cycles			
		1	2	3	4
IF=0.1	Entropy CB(%)	0.54	0.86	0.57	0.27
	KcenterGreedy CB(%)	0.84	0.44	0.77	0.77
	VAAL CB(%)	0.57	0.85	1.08	1.19
IF=0.3	Entropy CB(%)	0.40	1.11	0.31	0.08
	KcenterGreedy CB(%)	0.31	0.47	0.53	0.34
	VAAL CB(%)	0.28	0.24	0.42	0.91
IF=1	Entropy CB(%)	0.00	0.027	-0.15	-0.12
	KcenterGreedy CB(%)	0.19	0.14	0.03	0.05
	VAAL CB(%)	1.08	0.68	0.47	0.29

Table 4.1 – **Performance gain over AL baselines on CIFAR 10.**

Performance on CIFAR100. Fig. 4.6 presents the performance of the baselines and class-balanced counterparts on the CIFAR100 dataset. As can be seen in Fig. 4.6.a, the class balanced methods improve baselines marginally even though the dataset is balanced (IF=1). The improvements of class-balanced methods improve for the lower IF values (see Fig 4.6.b and c).

Notably, VAAL-CB achieves 3% improvement on average over the VAAL baseline in Fig. 4.6.b. See Table 4.2 for a detailed gain analysis. To put these improvements into perspective, the gain obtained by class balancing methods over the baselines is comparable to the improvement of those methods over Random. Specifically, Entropy-CB after 4 cycles achieves over 1% improvement over the Entropy baseline regardless of imbalance factor of the dataset.

Entropy L1 pseudo Label Fig. 4.7 presents the performance of another Entropy variation on CIFAR100 for comparison. "Entropy L1 Pseudo Label" benefits from "pseudo labels" defined as the most probable labels that the model assigns to unlabeled samples (the prediction of the model is then converted to a one-hot vector). This method utilizes the pseudo labels to balance the distribution of samples and select certain number of samples (specified by Ω in Eq.3) from each class with

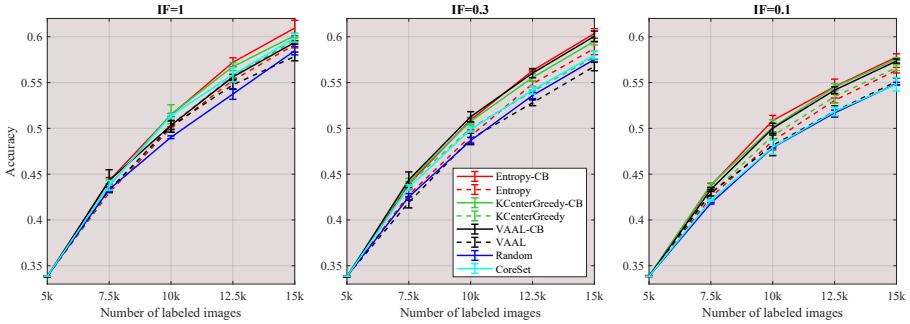


Figure 4.6 – **Performance evaluation.** Results for several active learning methods on CIFAR100 for different imbalance factors (IF).

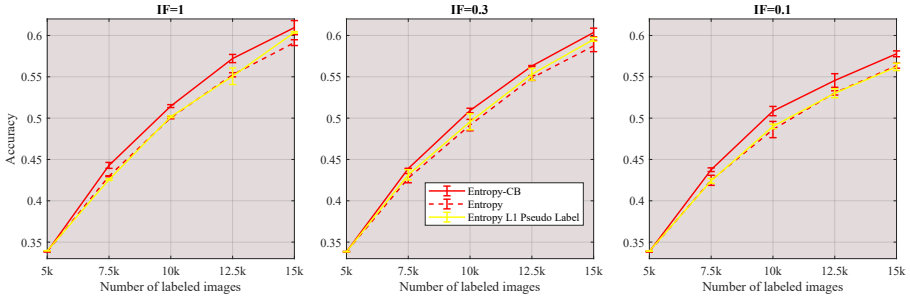


Figure 4.7 – **Performance evaluation.** Comparing Entropy standard, Entropy balanced by Pseudo Labels against the proposed Entropy CB.

maximum entropy. The experiments show that Entropy-CB outperforms Entropy L1 Pseudo Label both in terms of active learning performance (see Fig. 4.7) and the ability of class balancing (see λ tuning in Section 4).

Distribution of selected samples on CIFAR100 Fig. 4.8, 4.9 and 4.10 show the distribution of samples selected by AL methods on original (IF=1) and imbalanced (IF=0.3 and IF=0.1) CIFAR100 respectively. The L1 score above the distributions (introduced in Section 5.1) measures the $\ell 1$ distance from uniform distribution in the corresponding cycle. As can be seen, CB methods are remarkably effective in balancing the distribution of selected samples regardless of imbalance factor. It is worth mentioning in Fig 4.8 although the dataset is balanced, AL baselines (Entropy and KCenterGreedy) result in biased sampling. In contrast, CB methods provide

Imbalance Factor	Methods	Cycles			
		1	2	3	4
IF=0.1	Entropy CB(%)	1.28	2.23	1.50	1.43
	KcenterGreedy CB(%)	1.03	0.92	1.04	0.93
	VAAL CB(%)	0.37	1.86	2.32	2.23
IF=0.3	Entropy CB(%)	1.16	1.76	1.44	1.63
	KcenterGreedy CB(%)	0.28	0.76	1.52	1.70
	VAAL CB(%)	2.47	2.42	3.23	3.29
IF=1	Entropy CB(%)	1.37	1.40	1.96	1.82
	KcenterGreedy CB(%)	0.55	1.15	1.03	0.48
	VAAL CB(%)	1.01	0.11	0.86	1.53

Table 4.2 – **Performance gain over AL baselines on CIFAR 100.**

more balanced samples across all cycles and imbalance factors.

Performance on Tiny ImageNet. Tiny ImageNet is a challenging large scale dataset which we use to evaluate the scalability of our approach. Also to evaluate the generality of our approach we show the performance of class balancing applied to BALD as a Baysian approach[‡]. Table4.3 shows evidently the addition of class balancing to Entropy and BALD boost their performance on both balanced and imbalance datasets. Fig. 4.11 illustrates the performance of class balanced (CB) methods and AL baselines. As can be seen, both Entropy-CB and BALD-CB outperform the corresponding baselines. Notably in Tiny ImageNet, Random sampling serve as a competitive baseline. Nevertheless the addition of class balancing made Entropy-CB superior in almost all active learning cycles across different imbalance factors.

4.6 Conclusions

In this chapter, we have investigated the influence of class-imbalance on active learning performance. Class-imbalance can be caused by an imbalanced unlabeled data pool or by the sampling bias present in active learning algorithms. When

[‡]Representativeness-based methods are infeasible on large datasets.

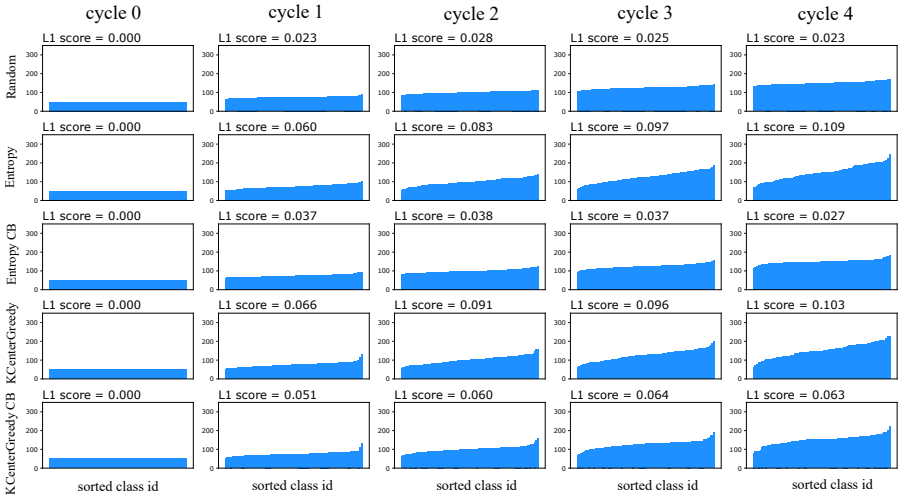


Figure 4.8 – **Distribution of samples selected by our proposed method (CB) compared to baselines on CIFAR100 with IF=1.**

aiming for good performance of the final classifier on all classes, class-imbalance has a detrimental effect. Therefore, to address the class-imbalance we proposed an optimization-based method that aims to balance classes. The method is general and can be combined with both the informativeness and representativeness criteria often used in active learning. Extensive experiments, on several existing datasets show that our method improves results of existing active learning methods. Our results suggests that class-balancing should be an important criteria when selecting samples, and that it should be considered next to the long-standing active learning criteria of informativeness and representativeness.

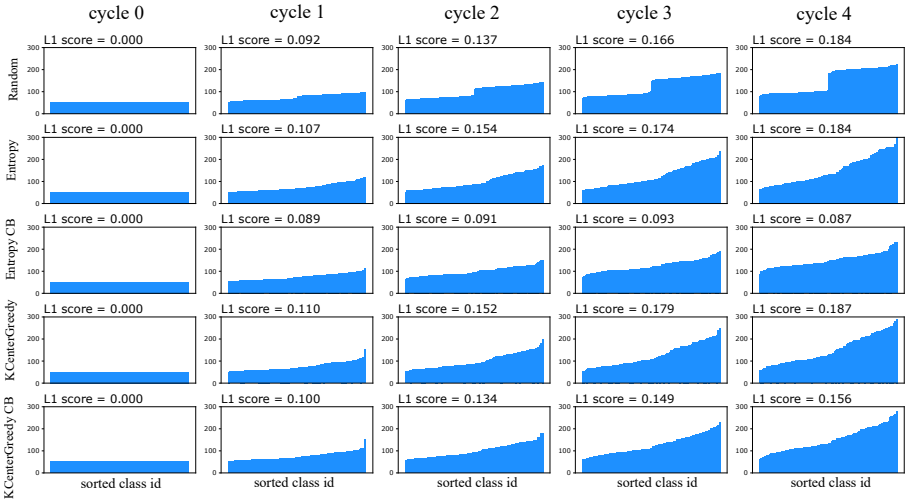


Figure 4.9 – **Distribution of samples selected by our proposed method (CB) compared to baselines on CIFAR100 with IF=0.3.**

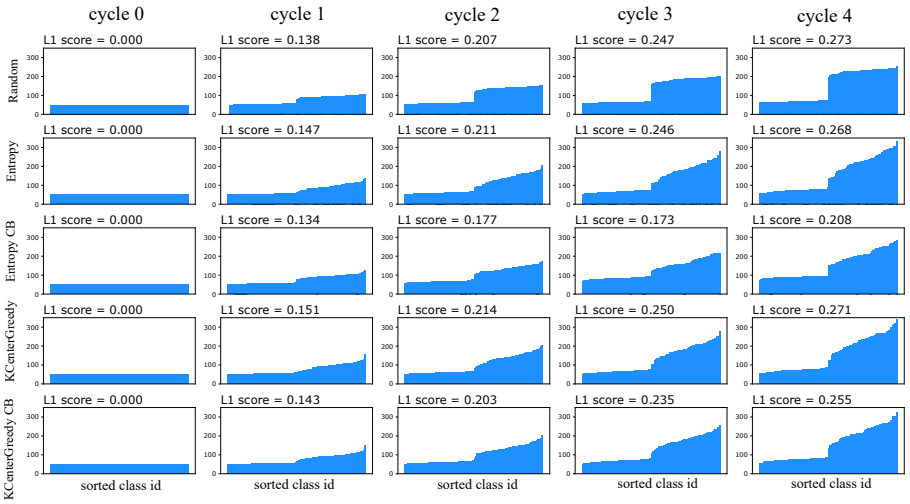


Figure 4.10 – **Distribution of samples selected by our proposed method (CB) compared to baselines on CIFAR100 with IF=0.1.**

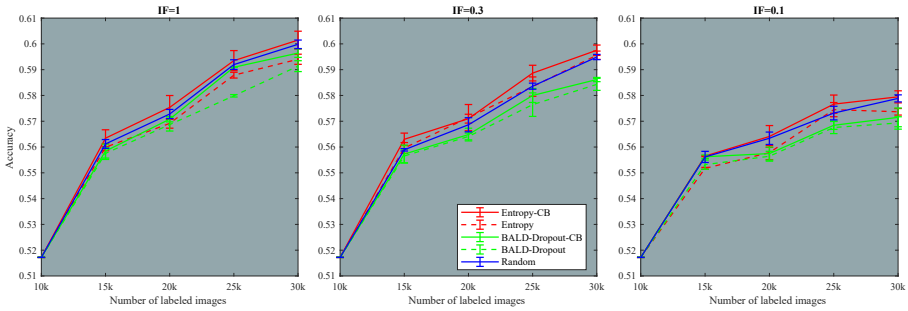


Figure 4.11 – **Performance evaluation.** Results for active learning methods on Tiny ImageNet with different imbalance factors (IF).

Imbalance Factor	Methods	Cycles			
		1	2	3	4
IF=0.1	Entropy CB (%)	0.48	0.63	0.21	0.58
	BALD CB(%)	0.31	0.10	0.08	0.21
IF=0.3	Entropy CB (%)	0.34	-0.04	0.52	0.19
	BALD CB(%)	0.07	0.07	0.36	0.19
IF=1	Entropy CB(%)	0.35	0.62	0.56	0.74
	BALD CB(%)	0.10	0.21	1.11	0.51

Table 4.3 – **Performance gain over AL baselines on Tiny ImageNet.**

5 Reducing Label Effort: Self-Supervised meets Active Learning *

Summary: *Active learning is a paradigm aimed at reducing the annotation effort by training the model on actively selected informative and/or representative samples. Another paradigm to reduce the annotation effort is self-training that learns from a large amount of unlabeled data in an unsupervised way and fine-tunes on few labeled samples. Recent developments in self-training have achieved very impressive results rivaling supervised learning on some datasets. The current work focuses on whether the two paradigms can benefit from each other. We studied object recognition datasets including CIFAR10, CIFAR100 and Tiny ImageNet with several labeling budgets for the evaluations. Our experiments reveal that self-training is remarkably more efficient than active learning at reducing the labeling effort, that for a low labeling budget, active learning offers no benefit to self-training, and finally that the combination of active learning and self-training is fruitful when the labeling budget is high. The performance gap between active learning trained either with self-training or from scratch diminishes as we approach to the point where almost half of the dataset is labeled.*

5.1 Introduction

Deep learning methods obtain excellent results on large annotated datasets [79]. However, labeling large amounts of data is labor-intensive and can be very costly. Therefore, the field of active learning explores algorithms that reduce the amount of labeled data that is required. This is achieved by labeling those unlabeled data samples (from the unlabeled data pool) that are considered most useful for the machine learning algorithm. The field of active learning can be roughly divided into two subfields. Informativeness-based methods aim to identify those data samples for which the algorithm is most uncertain [17, 50, 140]. Adding these samples to the labeled data pool is expected to improve the algorithm. Representativeness-

*This chapter is based on a publication in the IEEE/CVF International Conference of Computer Vision Workshops, 2021 [10]

based methods aim to label data in such a way that for all unlabeled data there is a 'representative' (defined based on distance in feature space) labeled sample [36, 115]. Active learning methods are typically evaluated by supervised training of the network on only the labeled data pool: the active learning method that obtains the best results, after a number of training cycles with a fixed label budget, is then considered superior.

Self-supervised learning of representation for visual data has seen stunning progress in recent years [19, 20, 21, 48, 55], with some unsupervised methods being able to learn representations that rival those learned supervised. The main progress has come from a recent set of works that learn representations that are invariant with respect to a set of distortions of the input data (such as cropping, applying blur, flipping, etc). In these methods, two distorted versions, called views, of the image are produced. Then the network is trained by enforcing the representations of the two views to be similar. To prevent these networks to converge to a trivial solution different approaches have been developed [48, 145]. The resulting representations are closing the gap with supervised-learned representation. For some downstream applications, such as segmentation and detection, the self-supervised representations even outperform the supervised representations [149].

As discussed, self-supervised learning can learn high-quality features that are almost at par with the features learned by supervised methods. As such it has greatly improved the usefulness of unlabeled data. The standard active learning paradigm trains an algorithm on the labeled data set, and based on the resulting algorithm selects data points that are expected to be most informative for the algorithm in better understanding the problem [117]. In this standard setup, the unlabeled data is not exploited to improve the algorithm. Given the huge performance gains that are reported by applying self-supervised learning, we propose to re-evaluate existing active learning algorithms in this new setting where the unlabeled data is exploited by employing self-supervised learning.

Self-supervised learning and active learning both aim to reduce the label-effort. Based on our experiments we conclude the following:

- In our evaluations on three datasets, Self-training is much more efficient than AL in reducing the labelling effort.

- Self-training + AL substantially outperforms AL methods. However, the performance gap diminishes for large labeling budget (approximately 50% of the dataset in our experiments).

- Based on results of three datasets, Self-training+AL marginally outperforms self-training but only when the labeling budget is high.

In general, our results suggest that self-supervised learning techniques are more efficient than active learning to reduce the label effort. A small additional boost can be obtained from active learning when reaching the high label budget.

This chapter is organized as follows: In section 5.2 we describe the related work. Next, in section 5.3 we introduce the proposed framework. Section 5.4 and 5.5 present the experimental setup and the evaluations on the datasets we used. Finally, section 5.6 discusses an interesting finding we observed in our work.

5.2 Related work

Active learning. Active Learning has been widely studied in various applications such as image classification [25, 43, 45], image retrieval [5], image captioning [32], object detection [155], and regression [42, 69].

Over the past two decades, several strategies have been proposed for sample query, which can be divided in three main categories: informativeness [8, 17, 44, 50, 140], representativeness [113, 115] and hybrid approaches [63, 138]. A comprehensive survey of these frameworks and a detailed discussion can be found in [117].

Among all the aforementioned strategies, the informativeness-based approaches are the most successful ones, with uncertainty being the most used selection criteria used in both bayesian [44] and non-bayesian frameworks [85]. In [44], they obtain uncertainty estimates through multiple forward passes with Monte Carlo Dropout, but it is computationally inefficient for recent large-scale learning as it requires dense dropout layers that drastically slow down the convergence speed. More recently, [3] measures the uncertainty of the model by estimating the expected gradient length. On the other hand, [83, 142] employ a loss module to learn the loss of a target model and select the images based on their output loss.

Representativeness-based methods rely on selecting examples by increasing diversity in a given batch [36]. The Core-set technique [115] selects the samples by minimizing the Euclidian distance between the query data and labeled samples in the feature space. The Core-set technique is shown to be an effective representation learning method, however, its performance is limited by the number of classes in the dataset. Furthermore, Core-set, like other distance-based approaches, are less effective due to feature representation in high-dimensional spaces since p-norms suffer from the curse of dimensionality [35]. In a different direction, [118] uses an adversarial approach for diversity-based sample query, which samples the data points based on the discriminator's output, seen as a selection criteria. Following the same strategy, improved versions have been proposed in [75, 146].

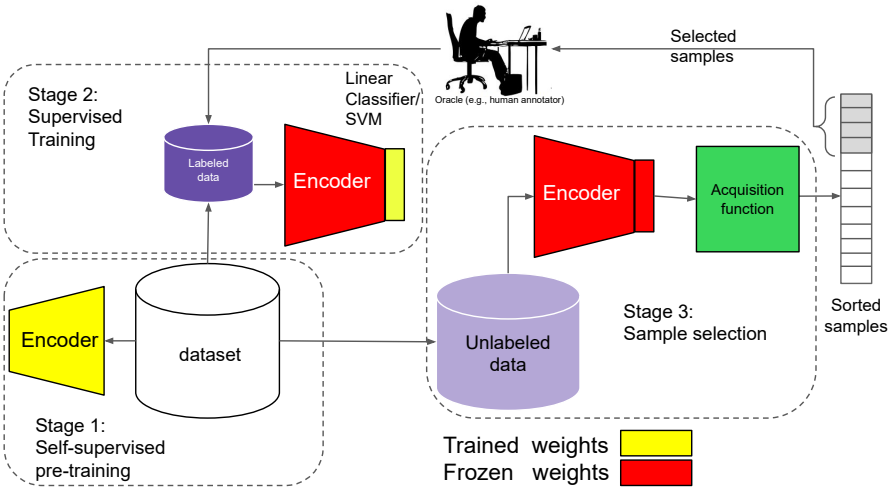


Figure 5.1 – **Overview of active learning framework enhanced by self supervised pre-training.** The framework consists of 3 stages: (i) Self supervised model is trained on the entire dataset. (ii) Given the frozen backbone and few labeled data, a linear classifier or an SVM is fine-tuned on top of the features in supervised way. (iii) Running the model as inference on the unlabeled data and sort the samples from least to highest informative/representative via acquisition function. Finally the top samples are queried to oracle for labeling and added to labeled set. Stages 2 & 3 are repeated until the total labeling budget finishes.

Self-supervised learning. In self-supervised learning, an auxiliary task is introduced. The data for this task should be readily available without the need for any human annotation. The auxiliary task allows to perform unsupervised learning and learn feature representations without the need of labels. Doersch et al. [34] introduce the task of estimating the relative position of image regions. Other examples include coloring gray-scale images [148], inpainting [102], and ranking [91].

In recent years, self-supervised learning has seen a significant performance jump with the introduction of contrastive learning [19], where representations are learned that are invariant with respect to several image distortions. Similar samples are created by augmenting an input image, while dissimilar are chosen by random. This connects to some extent unsupervised setting to previous contrastive methods used in metric learning [52, 133]. To make contrastive training more efficient MoCo method [55] and the improved version [20] use memory bank for learned embeddings what helps with an efficient sampling. This memory is kept in sync with the rest of the network during the training time by using a momentum

encoder. Approach named SwAV [18] use online clustering over the embedded samples. In this method negative exemplars are not defined. However, others cluster prototypes can play this role. Even more interesting are methods without any explicit contrastive pairs. BYOL [48] propose asymmetric network by introducing of an additional MLP predictor between two branches' outputs. One of the branch is keep "offline" - updated by a momentum encoder. SimSiam [21] goes even further and presents a simplified solution without a momentum encoder. It comparably good to other methods and does not need a big mini-batch size. A follow up work of BarlowTwins [145] proposes as simple solution as SimSiam with the use of a different loss function - a correlation based one for each pair in current training batch. Here, negatives are implicitly assumed to be in each mini-batch. No asymmetry is used in the network at all, but a bigger embedding size and mini-batches are proffered in comparison to SimSiam.

Previous works that integrated Active Learning and Self-supervised learning include [101, 154]. [154] proposes a query based graph AL method for datasets having structural relationships between the samples coming from few classes. In the context of exploration-driven agent, [101] uses Active Learning and Self-training to learn a policy that allows it to best navigate the environment space.

5.3 Preliminaries

The main objective of this chapter is to evaluate and compare the effectiveness of active learning when combined with recent advances in self-supervised learning. For this purpose we have developed a framework that comprises two parts: self supervised pre-training and active learning (see Figure 5.1). Primarily, we train the self supervised model as the pretrained model on the unlabeled samples. Next, we use an initial labeled data to finetune a linear classifier on top of pre-trained model. Then we run active learning cycles using the fine-tuned model to select the most informative and/or representative samples and query them for labeling. Hence the original dataset becomes partially labeled. We ablate the self-supervised and active learning components to study their benefits.

We start pretraining our model with SimSiam [21] self-supervised model. The model is based on siamese network trying to maximize the similarity between two augmentations of one image, subject to certain conditions for avoiding collapsing solutions. This enables us to obtain meaningful representations without using negative sample pairs. The rich representations could also potentially help the representative based active learning methods.

In the remainder of this section we describe the two components of the experimental framework in detail.

5.3.1 Active Learning

Given a large pool of unlabeled data $\mathscr{D}_{\mathscr{U}}$ and a total annotation budget B, the goal is to select b samples in each cycle to be annotated to maximize the performance of a classification model. In general, AL methods proceed sequentially by splitting the budget in several *cycles*. Here we consider the batch-mode variant [117], which annotates b samples per cycle, since this is the only feasible option for CNN training. At the beginning of each cycle, the model is trained on the labeled set of samples $\mathscr{D}_{\mathscr{L}}$. After training, the model is used to select a new set of samples to be annotated at the end of the cycle via an *acquisition function*. The selected samples are added to the labeled set $\mathscr{D}_L$ for the next cycle and the process is repeated until the annotation budget is spent. The acquisition function is the most crucial component and the main difference between AL methods in the literature. In the experiments we consider several acquisition functions including Informativeness [29] and Representativeness based methods [115, 118].

5.3.2 Self-supervised Learning

In this section, we shortly introduce self-supervised learning without contrastive sampling and more particularly SimSiam [21], the architecture we employ in this chapter.

For a given dataset $\mathscr{D}$, contrastive learning assumes sampling pairs of data points in order to create a good representation. Two main types of pairs are considered: *semantically similar pairs* (x, x^+) – provide an information about some form of close relation of data (based on labeled or unlabeled data); *negative pairs* (x, x^-) – in contrast to positives ones, two non-related samples are given. It is presumed that for a given x, x^- is dissimilar to x^+. Then, contrastive losses [52, 133] learn a new embedding space where a distance between positive pairs is smaller than negatives ones with some margin, e.g. $d(x, x^+) < d(x, x^-) + m$ for triplet loss [133]. That is a core of many metric learning methods [74, 96], where existing labels are used for a semantic similarity check.

Contrastive learning is also often applied in self-supervised learning methods. These methods aim to learn a semantically rich feature representation without the need of any labels. Different augmented views of the same image x form positive samples, while augmentation of different ones provide negatives. This is the base of SimCLR [19] method. However, it's shown that methods without explicit negative sampling prove competitive performance as well, e.g. SimSiam [21] or BYOL [48]. In such methods some additional architecture changes are usually applied, like using asymmetry with an additional predictor network as presented in Figure 5.2 for SimSiam.

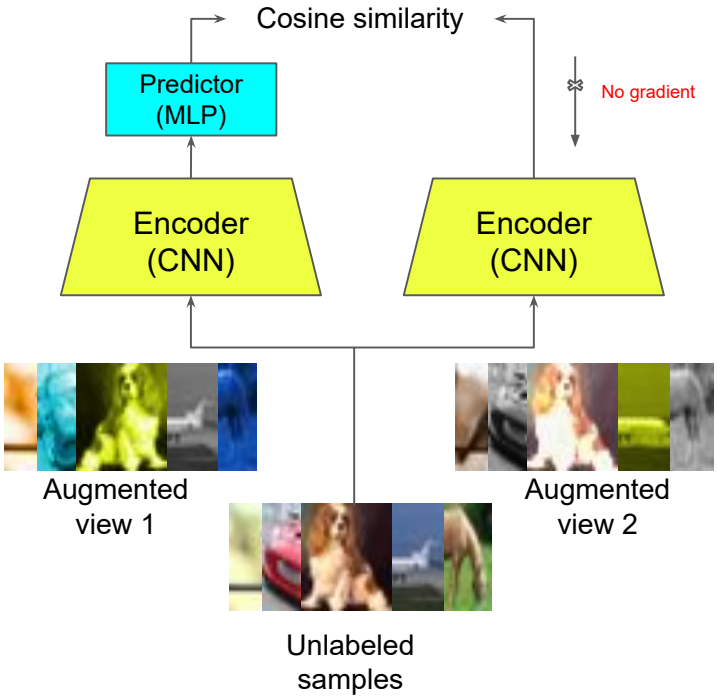


Figure 5.2 – **SimSiam architecture** Two augmented views of one image are processed by the same encoder network (a backbone plus a projection MLP). Then a prediction MLP is applied on one side, and a stop-gradient operation is applied on the other side. The model maximizes the similarity between both sides.

The main part is an encoder (CNN based network), learned end-to-end in an asymmetric Siamese architecture, where one branch got an additional predictor (MLP network) which outputs aims to be as close as possible to the other branch. The second branch is not updated in a backward propagation while training. For the similarity function a negative cosine distance is minimized given as:

$$\mathscr{L} = \mathscr{D}(p_1, z_2)/2 + \mathscr{D}(p_2, z_1)/2 \tag{5.1}$$

$$\mathscr{D}(p_1, z_2) = -\frac{p_1}{\|p_1\|_2} \cdot \frac{z_2}{\|z_2\|_2}, \tag{5.2}$$

where z_1, z_2 are encoded values respectively for $x1$ and $x2$ – two different augmented views of the same image x. p_1 and p_2 are encoded values additionally passed by a predictor network. There is no contrastive term in this approach, only the similarity is checked and enforced during learning. In SimSiam, besides simplicity, neither negatives mining nor large mini-batches are needed which significantly reduces the GPU requirements. This makes it a good fit for the evaluation proposed in this chapter.

5.4 Experimental Setup

To study the influence of the initial model, various amounts of initial labeled data and budget sizes are evaluated. For the initial labeled set, we considered 1%, 2% and 10% of the entire dataset that are uniformly selected from all classes at random. For one of the datasets we also evaluate 0.1% and 0.2% budget sizes. Before starting the active learning cycles we train the self-supervised model. Then we use the backbone as encoder from SimSiam architecture, freeze the weights and train a linear classifier or SVM on top of the backbone so we only finetune the last layer. At each cycle we start training either from scratch or, in case of self-training, we start from the pretrained self-supervised backbone. We train the model in c cycles until the total budget is exhausted. In each experiment the budget per cycle is equal to initial labeled set.

Datasets. To evaluate various methods, we use CIFAR10 and CIFAR100 [77] datasets with 50K images for training and 10K for testing. CIFAR10 and CIFAR100 have 10 and 100 object categories respectively and an image size of 32×32. To evaluate the scalability of the methods we evaluate on Tiny ImageNet dataset [80] with 90K images for training and 10K for testing. There are 200 object categories in Tiny ImageNet with an image size of 64×64.

Data Augmentation We use different augmentation policies for self supervised pre-training and supervised finetuning. [156] discusses how self-training outperforms normal pre-training in terms of stronger augmentation. For the self-training similar to [21] we used Geometric augmentations [135]: RandomResizedCrop with scale in [0:2; 1:0] and RandomHorizontalFlip. Color augmentation is ColorJitter with {brightness, contrast, saturation, hue} strength of {0.4, 0.4, 0.4, 0.1} with an applying probability of 0.8, and RandomGrayscale with an applying probability of 0.2. Blurring augmentation [19] has a Gaussian kernel with std in [0:1; 2:0]. For the supervised training we used the conventional RandomResizedCrop with scale [0.08, 1.0] and RandomHorizontalFlip.

Baselines. For the evaluation baselines we compared with Random sampling and several informative and representative-based approaches including Entropy sampling, KCenterGreedy, VAAL and SVM Min Margin. Below we describe the details of the methods we used.

Entropy [29] is an information theory measure that captures the average amount of information contained in the predictive distribution, attaining its maximum value when all classes are equally probable. Entropy sampling selects the most uncertain samples with highest entropy.

As a prominent representative method we evaluate *KCenterGreedy*, which is a greedy approximation of KCenter problem also known as min-max facility location problem [134]. The method selects samples having maximum distance from the nearest labeled samples in the embedding space. We compute the embeddings by running the self-trained model on unlabeled samples.

VAAL [118] is one of state-of-art methods that uses a variational autoencoder to map the distribution of labeled and unlabeled data to a latent space. A binary adversarial classifier is trained to predict if an image belongs to the labeled or the unlabeled pool. The unlabeled images which the discriminator classifies with lowest certainty as belonging to the labeled pool are considered to be the most representative. We used their official code and adapted them into our code to ensure an identical setting. To adapt VAAL for the self-training experiment we initialized and froze the backbone of the task learner.

SVM Min Margin [123] learns a linear SVM on the existing labeled data and chooses the samples that are closest to the decision boundary. To generalize SVM for the multi-class classification problem we adopt it by querying the samples that reside in margin area of decision boundaries.

Implementation details. Our method is implemented in PyTorch [100]. We train Resnet18 [57] that is widely used on CIFAR10 and CIFAR100 datasets. For the self-supervised training, the models are trained with SGD optimizer with momentum 0.9 and base learning rate of 0.03. As in [21] we train models for 800 epochs with batch-size of 512. We use a weight decay of 0.0001 for all parameter layers, including the BN scales and biases, in the SGD optimizer.

Given the pre-trained network, we train a supervised linear classifier on frozen features, which are from ResNet's global average pooling layer. The linear classifier training uses base lr=30 with a cosine decay schedule for 100 epochs, weight decay=0, momentum=0.9, batch size=256 with SGD optimizer.

To implement the SVM for the Min Margin method we used scikit learn python package [103] with linear kernel and set the regularization parameter to 5 in the experiments. To handle the multi-class problem, a one-vs-the-rest classification

scheme is chosen.

5.5 Experiments

To evaluate active learning methods we consider several scenarios in the initial labeled set and budget sizes. For the simplicity we refer to lower than 2% budget sizes as low budget regimes. In this section we inspect the contribution of self-supervised pre-training in active learning.

Performance on CIFAR10. Figure 5.3 shows active learning results on CIFAR10 dataset. The initial and per cycle budgets are 0.1%, 0.2%,1%, 2% and 10% of labeled data. The evaluated methods are divided into two groups: (i) methods using self-supervised pre-training represented by solid lines. (ii) Methods using models trained from scratch represented by dashed lines. As can be seen, self-training substantially improves all the sampling methods. In particular at the low budget regime, self-training drastically reduces the required labeling. Both types of methods achieve almost the full performance after labeling 50% of data that closes the gap between the self-supervised and supervised methods. The exact numbers are in Table 5.1. From the active learning perspective, Random sampling outperforms AL methods when the budget is less than 1%. However from 1% budget onward, AL + self-training methods transition to higher performance compared to Random sampling with self-training. For AL methods, trained from scratch, this transition happens after labeling 10% of data. Among AL methods with self-training, Entropy as informativeness method outperforms KCenterGreedy and VAAL. Note that the greatest active learning gain as a result of using self-training occurs after labeling 30% providing 20% less annotation that is equivalent to 10000 less labeling.

Performance on CIFAR100. Figure 5.4 presents active learning results on CIFAR100 dataset. The three set of curves correspond to three initial and per cycle budget sizes: 1%, 2% and 10%. Solid lines represent AL methods using self-supervised training. While dashed curves correspond to algorithms trained from scratch. As can be seen, self-training dramatically improves the methods without self training. In the low budget regime, self-training significantly reduces the required labeling. While AL methods w/o self-training achieve comparable performance to self-trained counterparts as we approach to 50% labeled data, meaning that the impact of self supervised pre-training diminishes when the budget increases. See Table 5.1 for detailed numbers. This can also be due to reaching almost the full performance. On CIFAR100, Random sampling outperforms Active learning methods under low budget regardless of using self-training. None of the studied methods foresee a

regime where the labeling budget is small, for example, labeling lower than 10%. Among the AL methods with self-training, representative-based methods perform better than Entropy as informative-based in low budget. On CIFAR100, the active learning gain of using self-training appeared almost after labeling 40% of dataset resulting in 10% less annotation that is equivalent to 5000 less labeling.

Performance on Tiny ImageNet. Tiny ImageNet is a challenging dataset in terms of diversity of classes. Active learning results on this dataset is presented in Figure 5.4. Similar to CIFAR100, the three set of curves correspond to 1%, 2% and 10% budget per cycle. Solid lines represent AL methods with self-supervised pre-training and dashed lines correspond to methods trained from scratch. As in other datasets, Self-training drastically reduces the required labeling in low budget scheme. As the labeling increases to 50% AL methods approach the performance of self-trained counterparts. However, unlike CIFAR datasets, AL methods require more than 50% labeling to close the performance gap they have from self-trained counterparts. Among the methods using self training, Random sampling shows superior performance. However, increasing labeled data reduces performance gap from the AL methods. For AL methods w/o self-training, the labeling budget is required to exceed 10% to improve upon Random sampling. In general, active learning fails to perform well under low budget regardless of using self-training. Again AL methods are not designed for low budget regime. Unless the model is trained from scratch with greater than 10% labeling budget, we observe no improvement with the usage of Active learning.

5.6 Discussion

The experiments in the previous section demonstrated that active learning methods enhanced by self-training do not work well in all budget schemes. However, it might be possible to estimate budgets above which the AL methods outperform Random sampling. Our experiments on three object recognition datasets show that there's a strong correlation (corr. coeff=0.99) between the number of samples per class required for AL and the number of classes in a datasets. Figure 5.6 presents the thresholds for the budget required for active learning to improve upon Random sampling when uses self-training. This is one interesting finding we observed which can provide a guideline based on the number of classes in a dataset to decide with a certain labeling budget whether it's beneficial to use active learning.

	Methods	Datasets	
		CIFAR10	CIFAR100
AL w/o Self-training	Entropy	0.908	0.646
	KCenterGreedy	0.895	0.641
AL + Self-training	Entropy	0.911	0.649
	SVM Min Margin	0.909	0.644
	VAAL	0.907	0.648
	KCenterGreedy	0.909	0.645

Table 5.1 – **Performance of AL methods with and without Self-training at** 50% **labeling.** For the high labeling budget, the gap between the performances of AL and AL+ Self-training is diminished.

5.7 Conclusions

This chapter analyzed active learning and self supervised approaches independently and unified to investigate how they can benefit from each other. Our experiments demonstrated that self-training is way more efficient than active learning at reducing the labeling effort. Besides, for a low labeling budget, active learning brings no benefit to self-training. Finally, the combination of active learning and self-training is beneficial only when the labeling budget is high. The performance gap between active learning with and without self-training diminishes as we approach to the point where almost half of the dataset is labeled.

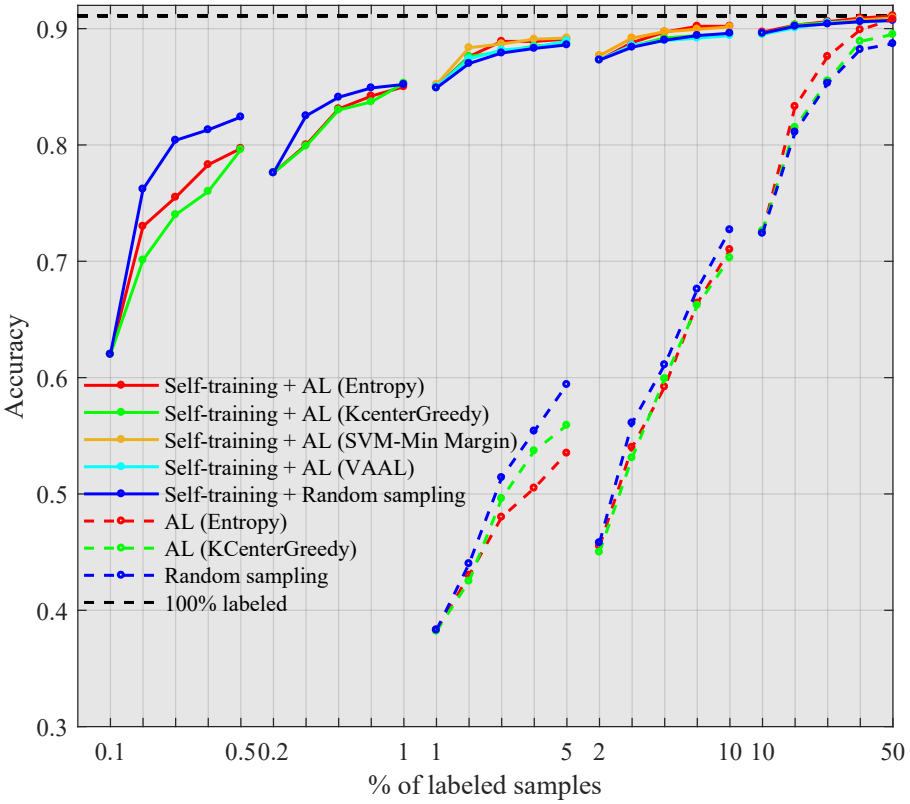


Figure 5.3 – **AL performance on cifar10** performance comparison between the addition of self-training to AL methods (solid lines) and AL methods (dashed lines). The initial and per cycle budget are equal in all the curves.

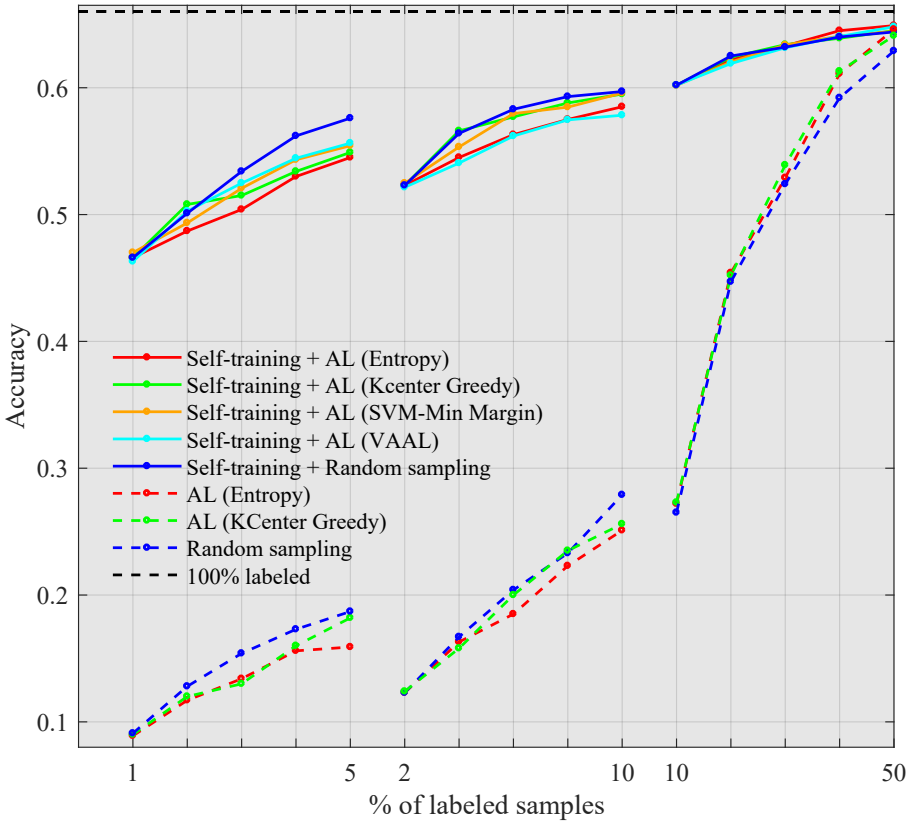


Figure 5.4 – **AL performance on cifar100** performance comparison between the addition of self-training to AL methods (solid lines) and AL methods (dashed lines). The initial and per cycle budget are equal in all the curves.

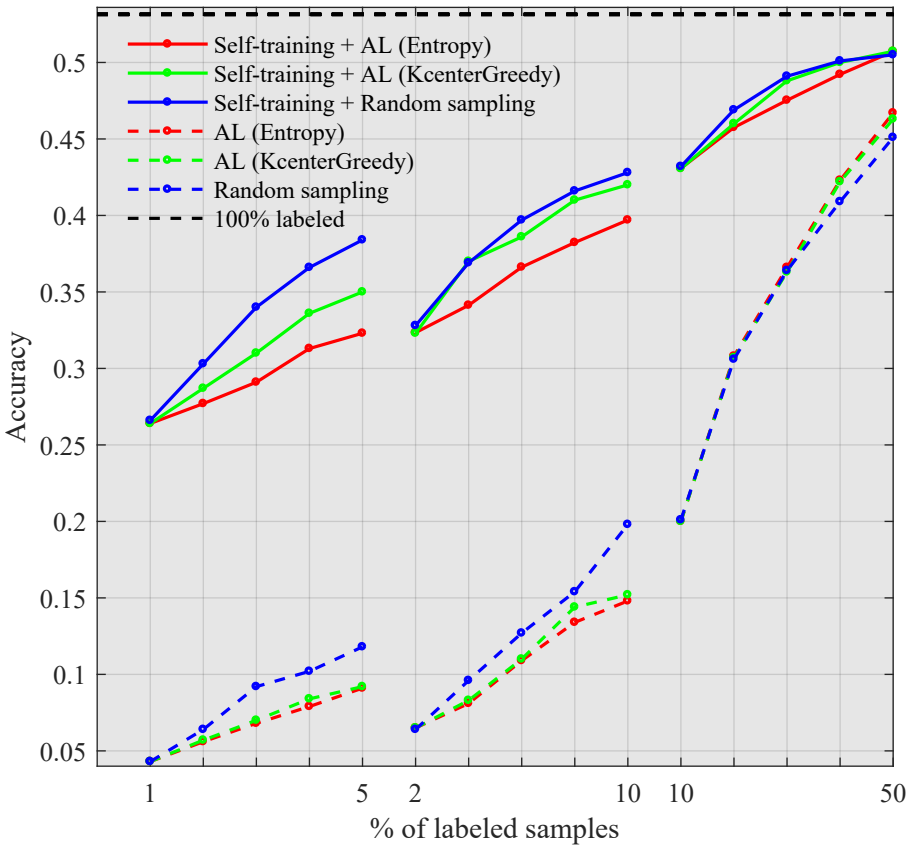


Figure 5.5 – **AL performance on Tiny ImageNet** performance comparison between the addition of self-training to AL methods (solid lines) and AL methods (dashed lines). The initial and per cycle budget are equal in all the curves.

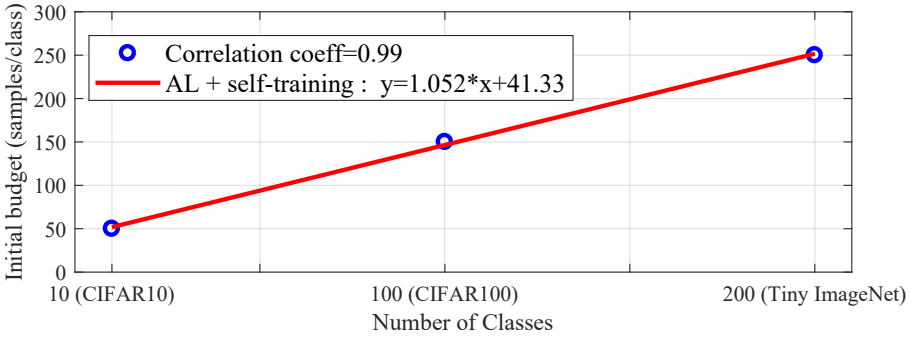


Figure 5.6 – **Correlation between number of samples per class required for AL and number of classes in the datasets.** Above these budgets, AL outperforms Random sampling in the self-supervised setting.

6 Conclusions and Future Work

6.1 Conclusions

Deep learning algorithms are very promising in many visual tasks where large amount of annotated data is available which are often difficult to obtain. In this thesis we studied important aspects of active learning methods with the aim of reducing the annotation effort.

In chapter 2, we introduced a novel active learning approach for object detection in videos which leverages the temporal coherence. We formulated our approach in terms of an energy minimization function of a graphical model built on tracked object detections. Additionally, we introduced a new synthetic dataset specially designed to evaluate active learning for object detection in the context of autonomous driving. Experimental results conducted on two datasets showed that our approach outperformed major active learning baselines.

In chapter 3, we proposed an active learning algorithm based on the learning dynamics of neural networks. We introduced the label-dispersion metric, which measures label-consistency during the training process. We showed that this measure obtains excellent results when used for active learning.

In chapter 4, we have investigated the influence of class-imbalance on active learning performance. Class-imbalance can be caused by an imbalanced unlabeled data pool or by the sampling bias present in active learning algorithms. To address the class-imbalance we proposed an optimization-based method that aims to balance classes. The method is general and can be combined with both the informativeness and representativeness criteria often used in active learning. Extensive experiments, on several existing datasets show that our method improves results of existing active learning methods. Our results suggests that class-balancing should be an important criteria when selecting samples, and that it should be considered next to the long-standing active learning criteria of informativeness and representativeness.

In chapter 5 we analyzed active learning and self supervised approaches independently and unified to investigate how they can benefit from each other. Our experiments demonstrated that self-training is way more efficient than active learning at reducing the labeling effort. Besides, for a low labeling budget, active learning

brings no benefit to self-training. Finally, the combination of active learning and self-training is beneficial only when the labeling budget is high.

6.2 Future work

A drawback of temporal coherence based active learning is that it is computationally more demanding than the baselines. We plan to minimize the computational overhead of our system in future research by solving the optimization problem with efficient algorithms.For future work, we are interested in exploring label-dispersion for other research fields such as out-of-distribution detection and within the context of lifelong learning. To further reduce the imbalance of distribution of training samples selected by active learning method we are interested to employ other distance metrics such as Wasserstien or L2 distances in our optimization problem. We are also willing to study the contribution of self-supervised learning in the active learning for the tasks of object detection and image segmentation. Other lines of research that we aim to pursue in the future are the application of Active Learning in unsupervised domain adaptation and multi-modal Active learning.

Summary of published works

1. **Javad Zolfaghari**, Abel Gonzalez-Garcia, Gabriel Villalonga, Bogdan Raducanu, Hamed Habibi Aghdam, Mikhail Mozerov, Antonio M López, Joost van de Weijer. Temporal coherence for active learning in videos. In proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2019.

2. Aymen Azaza, Joost van de Weijer, Ali Douik, **Javad Zolfaghari**, Marc Masana. Saliency from High-Level Semantic Image Features. Journal of SN Computer Science, 2020.

3. **Javad Zolfaghari**, Bogdan Raducanu, Joost van de Weijer. When Deep Learners Change Their Mind: Learning Dynamics for Active Learning. Proceedings of 19th International Conference on Computer Analysis of Images and Patterns (CAIP), 2021.

4. **Javad Zolfaghari**, Joost van de Weijer, Bartlomiej Twardowski, Bogdan Raducanu. Reducing Label Effort: Self-Supervised meets Active Learning. Proceeding of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021.

5. **Javad Zolfaghari**, Joost van de Weijer, Laura Lopez, Bogdan Raducanu. Class-Balanced Active Learning for Image Classification. Proceeding of the IEEE/CVF International Winter Conference on Applications of Computer Vision (WACV), 2022.

Bibliography

[1] Umang Aggarwal, Adrian Popescu, and Céline Hudelot. Active learning for imbalanced datasets. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1428–1437, 2020.

[2] Hamed H Aghdam, Abel Gonzalez-Garcia, Joost Van de Weijer, and Antonio M López. Active learning for deep detection neural networks. In *ICCV*, pages 3672–3680, 2019.

[3] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

[4] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

[5] Björn Barz, Christoph Käding, and Joachim Denzler. Information-theoretic active learning for content-based image retrieval. In *GCPR*, pages 650–666, 2018.

[6] William H. Beluch, Tim Genewein, Andreas Nurnberger, and Jan M. Kohler. The power of ensembles for active learning in image classification. In *CVPR*, pages 9368–9377, 2018.

[7] Javad Zolfaghari Bengar, Abel Gonzalez-Garcia, Gabriel Villalonga, Bogdan Raducanu, Hamed Habibi Aghdam, Mikhail Mozerov, Antonio M Lopez, and Joost Van de Weijer. Temporal coherence for active learning in videos. In *ICCV-W*, pages 914–923, 2019.

[8] Javad Zolfaghari Bengar, Bogdan Raducanu, and Joost van de Weijer. When deep learners change their mind: Learning dynamics for active learning. *arXiv preprint arXiv:2107.14707*, 2021.

[9] Javad Zolfaghari Bengar, Joost van de Weijer, Laura Lopez Fuentes, and Bogdan Raducanu. Class-balanced active learning for image classification. *arXiv preprint arXiv:2110.04543*, 2021.

[10] Javad Zolfaghari Bengar, Joost van de Weijer, Bartlomiej Twardowski, and Bogdan Raducanu. Reducing label effort: Self-supervised meets active learning. *arXiv preprint arXiv:2108.11458*, 2021.

[11] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision (ECCV)*, pages 850–865, 2016.

[12] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 49–56, 2009.

[13] Aditya R Bhattacharya, Ji Liu, and Shayok Chakraborty. A generic active learning framework for class imbalance applications. In *BMVC*, page 121, 2019.

[14] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 26(9):1124–1137, 2004.

[15] Clemens-Alexander Brust, Christoph Käding, and Joachim Denzler. Active learning for deep object detection. In *VISAPP*, 2019.

[16] Mausam C Lin. Active learning with unbalanced classes & example-generated queries. In *AAAI Conference on Human Computation*, 2018.

[17] Wenbin Cai, Ya Zhang, Siyuan Zhou, Wenquan Wang, Chris Ding, and Xiao Gu. Active learning for support vector machines with maximum model change. In *Machine Learning and Knowledge Discovery in Databases*, pages 211–226. Springer, 2014.

[18] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

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

[20] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[21] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[22] Kashyap Chitta, Jose M Alvarez, and Adam Lesnikowski. Large-scale visual active learning with deep probabilistic ensembles. *arXiv preprint arXiv:1811.03575*, 2018.

[23] Kashyap Chitta, Jose M. Alvarez, and Adam Lesnikowski. Large-scale visual active learning with deep probabilistic ensembles. *arXiv preprint arXiv:1811.03575v3*, 2019.

[24] Jongwon Choi, Kwang Moo Yi, Jihoon Kim, Jinho Choo, Byoungjip Kim, Jinyeop Chang, Youngjune Gwon, and Hyung Jin Chang. Vab-al: Incorporating class imbalance and difficulty with variational bayes for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6749–6758, 2021.

[25] Jongwon Choi, Kwang Moo Yi, Jihoon Kim, Jinho Choo, Byoungjip Kim, Jinyeop Chang, Youngjune Gwon, and Hyung Jin Chang. Vab-al: Incorporating class imbalance and difficulty with variational bayes for active learning. In *CVPR2021*, pages 6749–6758, 2021.

[26] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[27] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, pages 9268–9277, 2019.

[28] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR*, pages 4109–4118, 2018.

[29] Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier, 1995.

[30] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *ICML*, pages 208–215, 2008.

[31] Cheng Deng, Xianglong Liu, Chao Li, and Dacheng Tao. Active multi-kernel domain adaptation for hyperspectral image classification. *Pattern Recognition*, 77:306–315, 2018.

[32] Yue Deng, KaWai Chen, Yilin Shen, and Hongxia Jin. Adversarial active learning for sequence labeling and generation. In *IJCAI*, pages 4012–4018, 2018.

[33] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *JMLR*, 17(83):1–5, 2016.

[34] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[35] David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000.

[36] Suyog Dutt Jain and Kristen Grauman. Active image segmentation propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2864–2873, 2016.

[37] Ehsan Elhamifar, Guillermo Sapiro, Allen Yang, and S Shankar Sasrty. A convex optimization framework for active learning. In *ICCV*, pages 209–216, 2013.

[38] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[39] Sebastian Farquhar, Yarin Gal, and Tom Rainforth. On statistical bias in active learning: How and when to fix it. *arXiv preprint arXiv:2101.11665*, 2021.

[40] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 32(9):1627–1645, 2010.

[41] L.C. Freeman. *Elementary applied statistics: for students in behavioral science*. Wiley, 1965.

[42] A. Freytag, E. Rodner, and J. Denzler. Selecting influential examples: Active learning with expected model output changes. In *European Conference on Computer Vision (ECCV)*, pages 562–577, 2014.

[43] Weijie Fu, Meng Wang, Shijie Hao, and Xindong Wu. Scalable active learning by approximated error reduction. In *KDD*, pages 1396–1405, 2018.

[44] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *ICML*, pages 1183–1192, 2017.

[45] E. Gavves, T. E. J. Mensink, T. Tommasi, and T Snoek, C. G. M.and Tuytelaars. Active transfer learning with zero-shot priors: Reusing past datasets for future tasks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–9, 2015.

[46] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32(11):1231–1237, 2013.

[47] S. Alireza Golestaneh and Kris M. Kitani. Importance of self-consistency in active learning for semantic segmentation. In *BMVC*, 2020.

[48] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[49] Quanquan Gu, Tong Zhang Zhang, Chris Ding, and Jiawei Han. Selective labeling via error bound minimization. In *NIPS*, pages 1–9, 2012.

[50] Yuhong Guo. Active instance sampling via matrix partition. In *NIPS*, pages 1–9, 2010.

[51] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2021.

[52] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[53] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S Huang. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465*, 2016.

[54] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[55] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[56] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[58] Fabian Caba Heilbron, Joon-Young Lee, Hailin Jin, and Bernard Ghanem. What do i annotate next? an empirical study of active learning for action localization. In *ECCV*, pages 212–229, 2018.

[59] Judy Hoffman, Sergio Guadarrama, Eric Tzeng, Jeff Donahue, Ross B. Girshick, Trevor Darrell, and Kate Saenko. Lsda: Large scale detection through adaptation. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1–9, 2014.

[60] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, pages 5375–5384, 2016.

[61] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7310–7311, 2017.

[62] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. *IEEE TPAMI*, 36(10):1936–1949, 2014.

[63] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. *IEEE Trans. on PAMI*, 10(36):1936–1949, 2014.

[64] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7610–7619, 2020.

[65] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proc. of the Int'l Conf. on Artificial Intelligence*, volume 56, pages 111–117, 2000.

[66] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. Self-paced curriculum learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[67] SouYoung Jin, Aruni RoyChowdhury, Huaizu Jiang, Ashish Singh, Aditya Prasad, Deep Chakraborty, and Erik Learned-Miller. Unsupervised hard example mining from videos for improved object detection. In *European Conference on Computer Vision (ECCV)*, pages 307–324, 2018.

[68] Ajay J. Joshi, Fatih Porikli, and Nikolaos P. Papanikolopoulos. Scalable active learning for multiclass image classification. *IEEE Trans. on PAMI*, 34(11):2259–2273, 2012.

[69] Christoph Käding, Erik Rodner, Alexander Freytag, Oliver Mothes, Björn Barz, and Joachim Denzler. Active learning for regression tasks with expected model output changes. In *BMVC*, pages 1–15, 2018.

[70] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. Object detection in videos with tubelet proposal networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 727–735, 2017.

[71] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE TCSVT*, 28(10):2896–2907, 2018.

[72] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Active learning with gaussian processes for object categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[73] Vasiliy Karasev, Avinash Ravichandran, and Stefano Soatto. Active frame, location, and detector selection for automated and manual video annotation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2131–2138, 2014.

[74] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[75] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *CVPR*, pages 8166–8175, 2021.

[76] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimizedvia graph cuts? *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 26(2):147–159, 2004.

[77] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. PhD thesis, University of Toronto, 2012.

[78] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

[79] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[80] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7:7, 2015.

[81] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[82] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR*, pages 3–12, 1994.

[83] Minghan Li, Xialei Liu, Joost van de Weijer, and Bogdan Raducanu. Learning to rank for active learning: A listwise approach. In *ICPR*, pages 5587–5594, 2020.

[84] Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *cvpr*, pages 860–866, 2013.

[85] Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *CVPR*, pages 859–866, 2013.

[86] Xin Li and Yuhong Guo. Multi-level adaptive active learning for scene classification. In *European Conference on Computer Vision (ECCV)*, pages 234–249, 2014.

[87] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[88] X. Lin and D. Parikh. Active learning for visual question answering: An empirical study. *arXiv preprint arXiv:1711.01732*, 2017.

[89] Mason Liu and Menglong Zhu. Mobile video object detection with temporally-aware feature maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5686–5695, 2018.

[90] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, pages 21–37, 2016.

[91] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Exploiting unlabeled data in cnns by self-supervised learning to rank. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1862–1878, 2019.

[92] David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

[93] Vashisht Madhavan and Trevor Darrell. The bdd-nexar collective: A large-scale, crowsourced, dataset of driving scenes. Master's thesis, EECS Department, University of California, Berkeley, 2017.

[94] Dwarikanath Mahapatra, Behzad Bozorgtabar, Jean-Philippe Thiran, and Mauricio Reyes. Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 580–588. Springer, 2018.

[95] Christoph Mayer and Radu Timofte. Adversarial sampling for active learning. In *WACV*, pages 3071–3079, 2020.

[96] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020.

[97] Mark EJ Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[98] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017.

[99] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

[100] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[101] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pages 5062–5071. PMLR, 2019.

[102] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

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

[104] Junran Peng, Xingyuan Bu, Ming Sun, Zhaoxiang Zhang, Tieniu Tan, and Junjie Yan. Large-scale object detection in the wild from imbalanced multi-labels. In *CVPR*, pages 9709–9718, 2020.

[105] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 4334–4343, 2018.

[106] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *arXiv preprint arXiv:2009.00236*, 2020.

[107] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[108] G. Ros, L. Sellart, J. Materzyska, D. Vázquez, and A.M. López. The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016.

[109] Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv preprint arXiv:1808.03305*, 2018.

[110] Soumya Roy, Asim Unmesh, and Vinay P. Namboodiri. Deep active learning for object detection. In *BMVA British Machine Vision Conference (BMVC)*, pages 1–12, 2018.

[111] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[112] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[113] P.T. Saito, C.T. Suzuki, J.F. Gomes, P.J. de Rezende, and A.X. Falcão. Robust active learning for the diagnosis of parasites. *Pattern Recognition*, 48(11):3572–3583, 2015.

[114] Andrew I. Schein and Lyle H. Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265, 2007.

[115] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, pages 1–13, 2018.

[116] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[117] Burr Settles. *Active learning*. Morgan Claypool, 2012.

[118] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *ICCV*, 2019.

[119] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2213–2222, 2017.

[120] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[121] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2019.

[122] Simon Tong. *Active learning: theory and applications*. Stanford University, 2001.

[123] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[124] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[125] Alexander Vezhnevets, Joachim M. Buhmann, and Vittorio Ferrari. Active learning for semantic segmentation with expected change. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3162–3169, 2012.

[126] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *IJCV*, 108(1–2):97–114, 2014.

[127] Jun Wang, Shaoguo Wen, Kaixing Chen, Jianghua Yu, Xin Zhou, Peng Gao, Changsheng Li, and Guotong Xie. Semi-supervised active learning for instance segmentation via scoring predictions. In *Proc. of BMVC*, 2020.

[128] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016.

[129] Shiyao Wang, Yucong Zhou, Junjie Yan, and Zhidong Deng. Fully motion-aware network for video object detection. In *European Conference on Computer Vision (ECCV)*, pages 542–557, 2018.

[130] Shuo Wang, Yuexiang Li, Kai Ma, Ruhui Ma, Haibing Guan, and Yefeng Zheng. Dual adversarial network for deep active learning. In *ECCV*, pages 1–17, 2020.

[131] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense r-cnn. In *CVPR*, pages 10760–10770, 2020.

[132] Xinyue Wang, Bo Liu, Siyu Cao, Liping Jing, and Jian Yu. Important sampling based active learning for imbalance classification. *Science China Information Sciences*, 63(8):1–14, 2020.

[133] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.

[134] Gert W Wolf. Facility location: concepts, models, algorithms and case studies. *International Journal of Geographical Information Science*, 25(2):331–333, 2011.

[135] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[136] Yan Yan, Feiping Nie, Wen Li, Chenqiang Gao, Yi Yang, and Dong Xu. Image classification by cross-media active learning with privileged information. *IEEE Trans. on Multimedia*, 18(12):2494–2502, 2016.

[137] Yazhou Yang and Marco Loog. A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, 83:401–415, 2018.

[138] Yazhou Yang and Marco Loog. A variance maximization criterion for active learning. *Pattern Recognition*, 78:358–370, 2018.

[139] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G. Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *IJCV*, 113(2):113–127, 2015.

[140] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *IJCV*, 113(2):113–127, 2015.

[141] Angela Yao, Juergen Gall Gall, Christian Leistner, and Luc Van Gool. Interactive object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3242–3249, 2012.

[142] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 93–102, 2019.

[143] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

[144] Hualong Yu, Xibei Yang, Shang Zheng, and Changyin Sun. Active learning from imbalanced data: A solution of online weighted extreme learning machine. *IEEE transactions on neural networks and learning systems*, 30(4):1088–1103, 2018.

[145] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

[146] Beichen Zhang, Liang Li, Shijie Yang, Shuhui Wang, Zheng-Jun Zha, and Qinming Huang. State-relabeling adversarial active learning. In *CVPR*, pages 8756–8765, 2020.

[147] Dan Zhang, Fei Wang, Zhenwei Shi, and Changshui Zhang. Interactive localized content based image retrieval with multiple-instance active learning. *Pattern Recognition*, 43(2):478–484, 2010.

[148] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[149] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020.

[150] Qian Zhao, Deyu Meng, Lu Jiang, Qi Xie, Zongben Xu, and Alexander G Hauptmann. Self-paced learning for matrix factorization. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[151] Jia-Jie Zhu and José Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017.

[152] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, pages 408–417, 2017.

[153] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2349–2358, 2017.

[154] Yanqiao Zhu, Weizhi Xu, Qiang Liu, and Shu Wu. When contrastive learning meets active learning: A novel graph active learning paradigm with self-supervision. *arXiv preprint arXiv:2010.16091*, 2020.

[155] Javad Zolfaghari Bengar, Abel Gonzalez-Garcia, Gabriel Villalonga, Bogdan Raducanu, Hamed Habibi Aghdam, Mikhail Mozerov, Antonio M López, and Joost van de Weijer. Temporal coherence for active learning in videos. In *ICCV Workshops*, 2019.

[156] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*, 2020.