université PARIS-SACLAY

Active and Incremental Deep learning with class imbalanced data

Apprentissage profond actif et incrémental avec des données de classes déséquilibrées

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 573 : interfaces : matériaux, systèmes, usages (INTERFACES)
Spécialité de doctorat : Informatique
Graduate School : Sciences de l'ingénierie et des systèmes, Référent : CentraleSupélec

Thèse préparée dans les unités de recherche Mathématiques et Informatique pour la Complexité et les Systèmes (Université Paris-Saclay, CentraleSupélec) et Institut LIST (Université Paris-Saclay, CEA) , sous la direction de Céline HUDELOT, professeure, et le co-encadrement de Adrian POPESCU, docteur.

Thèse soutenue à Paris-Saclay, le 25 Mars 2022, par

Umang AGGARWAL

Composition du jury

Nicolas Thome Professeur, CNAM	Président
Hanène Azzag Maîtresse de conférences, Université Paris 13	Rapportrice & Examinatrice
Ioannis Kannelos Professeur, IMT Atlantique	Rapporteur & Examinateur
Florian Yger Maître de conférences, Université Paris-Dauphine	Examinateur
Céline Hudelot Professeur, Université Paris-Saclay	Directrice de thèse
Adrian Popescu Ingénieur-Chercheur, CEA-LIST	Co-encadrament de thèse

THESE DE DOCTORAT
NNT : 2022UPAST027

université PARIS-SACLAY

ÉCOLE DOCTORALE
Interfaces:
matériaux, systèmes, usages

Titre : Apprentissage profond actif et incrémental avec des données de classes déséquilibrées
Mots clés : déséquilibre des classes, apprentissage actif, apprentissage incrémentiel (version en français)

Résumé : Les approches d'apprentissage profond sont très performantes aujourd'hui dans un large éventail de problèmes et en particulier pour les tâches de reconnaissance visuelle. Bien que de grands ensembles de données, parfaitement étiquetées, soient disponibles à des fins académiques, les jeux de données du monde réel présentent plusieurs problèmes tels que le non-équilibre des classes, leur coût d'annotation ainsi que leur caractère dynamique dans un grand nombre d'applications. Dans cette thèse, nous nous interessons aux limitations des approches d'apprentissage profond supervisées dans ce contexte. Notre proposition est d'exploiter des schémas d'apprentissage itératifs profonds tels que l'apprentissage actif et l'apprentissage incrémental, tout en tenant compte de la nature non-équilibrée des ensembles de données du monde réel.

Dans notre première contribution, nous considérons le cas difficile du démarrage à froid dans lequel nous ne disposons pas de jeu de données initial labélisé permettant d'initier le processus d'apprentissage actif. Nous proposons donc un schéma d' apprentissage actif en une seule étape dans lequel un modèle source est réutilisé pour la sélection d'instances à annoter dans l'ensemble de données non supervisées du domaine cible.

Dans notre deuxième contribution, nous supposons la disponibilité d'un jeu de données étiquetées initial dans le domaine cible et nous proposons des solutions qui suit le cadre itératif classique dans lequel les échantillons sont progressivement annotés pour mettre à jour le modèle appris. Pour le deuxième cas, nous nous placons dans un cadre d'apprentissage incremental déséquilibré en se concentrant sur les méthodes de calibration dont l'objectif est de réduire le biais de prédiction entre les classes majoritaires et minoritaires. Nous concluons que le problème d'apprentissage incrémental avec une mémoire à budget fixe pour les classes précédemment apprises peut être traité efficacement comme un problème d'apprentissage de déséquilibre de classe.

Title : Active and Incremental Deep learning with class imbalanced data
Keywords : class imbalance, active learning, incremental learning (version en anglais)

Abstract : Deep learning approaches have been successful in a large range of problems and in particular for visual recognition tasks. Though large and perfect labeled datasets are available for academic research, the real world application datasets have several issues such as class imbalance, annotation cost as well as dynamic nature of the dataset in some applications. In our work, we perform a joint study of these issues to tackle a more practical scenario. Our proposition is to leverage deep iterative learning schemes such as active learning and incremental learning while taking into account the imbalance nature of real-world datasets.

Our first contributions are tailored from the active learning, where a hard cold start setting is considered in which no initial labeled setting is available. We thus propose a single stage active learning scheme in which a good embedding model is used in the selection of instances to annotate in the unsupervised dataset of the target domain. In our second contribution, we assume the availability of an initial labeled dataset in the target domain (soft cold start problem) and we propose solutions in the classical iterative setting in which samples are progressively annotated to update the learned model.

In the class incremental setting, we perform a detailed study of imbalanced incremental learning with focus on calibration methods whose objective is to reduce the prediction bias between majority and minority classes. We conclude that the incremental learning problem with a fixed-budget memory for the previously learnt classes can be effectively treated as a class imbalance learning problem.

Acknowledgement

This thesis was carried out at Laboratoire d'Analyse Sémantique Texte Image (LASTI) of Commissariat l'Énergie Atomique (CEA) and Laboratoire Mathématiques et Informatique pour la Complexité et les Systèmes (MICS) of Centrale Supélec, both members of University Paris-Saclay.

Firstly, I would like to thank my supervisors Céline Hudelot and Adrian Popescu for their patience, kindness and availability for all the duration of the thesis. This thesis would not have been completed without their invaluable support.

I would also like to thank my colleagues, particularly Fréjus and Omar for their valuable inputs and help during the thesis.

Finally, I would like to thank my family and friends for all their love and support.

Table des matières

1 - Introduction

1.1 . Context : visual recognition at the era of deep learning

Deep learning algorithms and in particular supervised deep neural models have allowed impressive progress in the last decade for various visual recognition tasks such as classification, object detection or semantic segmentation (see Figure 1.1 for a quick overview on these tasks). Indeed, for all of these tasks, performances, evaluated on public benchmarks, have taken a step forward thanks to deep neural models. For instance, for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [194], the average top1 accuracy has increased from 50.9% using pre-deep learning models in 2011 to 90.2% in recent work [53]. Similar gain is seen for object detection task on MS-COCO [142], where the average precision of bounding boxes has increased from 34.9% in 2015 with Faster R-CNN models [66] to 58.7% in recent method [148]. In particular, deep learning has shifted the paradigm from using hand-crafted features to **representation learning** with multi-layered models [16]. Moreover, deep neural networks (DNNs) have proved to be efficient in learning powerful hierarchical data representations that can even be **transferred** to other tasks [170].

The **advent of Graphics Processing Units (GPUs)** [34, 232] to match the computationally intensive nature of deep learning algorithms, the better design of **deep learning architectures** [125, 39, 206, 66, 89] and the availability of **large annotated datasets** [193, 166, 142] are some of the major factors which explain performance gains and ubiquity of deep learning. For instance, Convolutional Neural Networks (CNNs), which provide the basic architecture design for most computer vision tasks today were envisioned as early as 1988 for phenome classification task [6]. In 1989, Yann LeCun used CNNs for hand-written character recognition and trained a neural network using backpropagation algorithm [131]. The usability of convolutional neural networks was limited at that time by the computational hardware available and led to a hiatus between the work of LeCun and the resurgence of CNNs in the last decade with the use of specialized GPUs hardware. GPUs which were initially developed for gaming consoles effectively performed the repetitive computations required in neural nets and thus helped to resolve the hardware bottleneck in DNNs [172] and opened the era of deep neural networks.

In particular, GPUs allowed advances in architecture design with more trainable parameters resulting in an increased representation ability of deep neural networks. As shown in Figure 1.2, it exists a strong relation between the size of the deep models (number of parameters) and their efficiency in term of task accuracy. Today, deep learning architectures include millions of parameters which have to be optimised for any given task. This overparameterized nature of deep models is

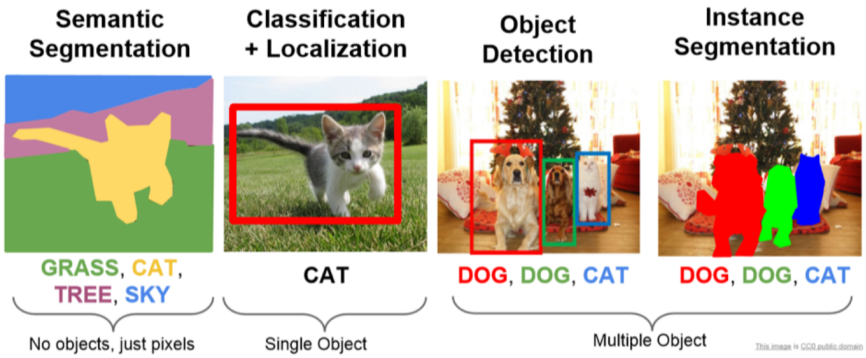


Figure 1.1 – Illustration of various tasks in Computer vision for which deep neural networks provide the state of the art performances [136].

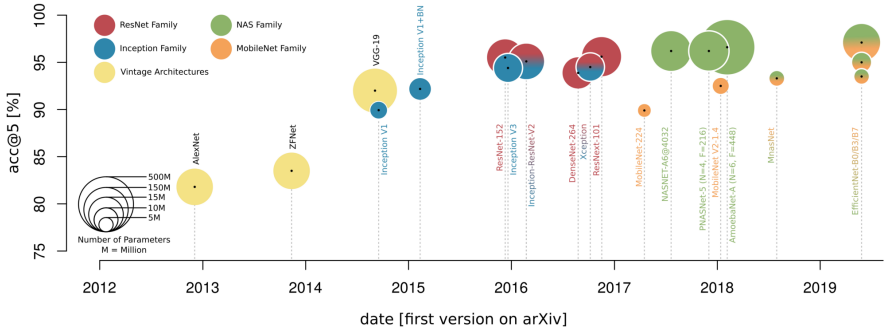


Figure 1.2 – Evolution of model sizes and accuracy on ILSVRC [95]. The model sizes have increased with time along with a corresponding increase in model accuracy.

an asset for learning complex representations, but is also one of the limitations of these models, since it limits their interpretability [143]. Indeed, while some progress has been made towards increasing the interpretability of deep learning models [46], a trade-off between interpretability and performance has been established due to overparameterized nature of large deep learning models [72].

Another major factor that explains the rise in use of deep learning models is the increase of available datasets and benchmarks. For research purposes, the computer vision community has developed numerous large and high-quality annotated datasets such as ImageNet [193] and CIFAR [124] for the task of object recognition, MS COCO [142] and Open Images [128] for object detection and segmentation as well as multi-label classification or Google landmarks [166] and indoor scene recognition [253] for the task of scene understanding among others. Annotated datasets are build using strong and rich lexical resources which are provided by

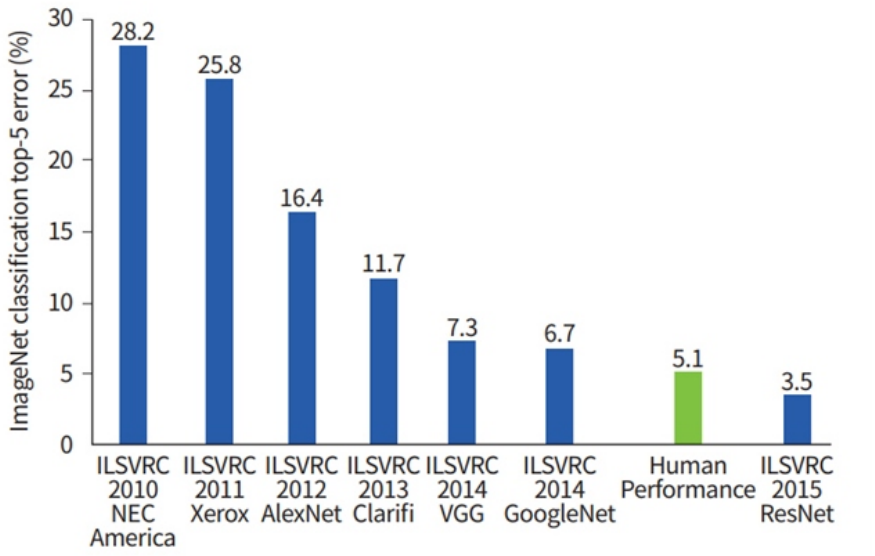


Figure 1.3 – Top-5 Error rate of yearly winners on ILSVRC [145] classification task. ResNet architecture surpassed the human performance of 5.1% in the year 2015.

human supervision during the annotation process. Therefore, building these datasets are costly on both time and human resources. For instance, the ImageNet database is build on the well known Wordnet lexical resource [157] that encodes common sense knowledge. Each of the 14 million images of the ImageNet database is assigned to one among of 22000 classes by a process that involved costly manual annotation [193].

These datasets have served the community to benchmark and develop different approaches. For instance, the long history between deep models and ILSVRC dataset is shown in Figure 1.3 [145]. While these academic datasets have allowed deep learning to advance on several difficult tasks, the availability of large perfectly annotated datasets in most practical scenarios can not be assumed. This is a major bottleneck for real-world applications as the performance of deep learning models is highly linked to the training dataset size. Authors in [211] show that for computer vision tasks, performance increases logarithmically to the size of the dataset. Also, as the size of the deep models has grown to improve its representational ability, larger datasets are needed to avoid over-fitting where the model memorizes the training data instead of learning useful patterns from the data [149]. Thus, while the increase in model and dataset sizes has enabled deep learning models to vastly outperform traditional machine learning models, designing deep learning solutions has become highly dependent on the availability of large datasets, generating some

important limitations.

1.2 . Data-dependent deep neural models : some limitations

Despite its success, deep learning has several potential and important drawbacks arising from its data dependent nature, in particular in the context of the design and the deployment of deep models for real case applications. We details these factors in the following.

- **Annotating data is a very expensive task** While large scale image collections are now widely available, for instance on the Web-corpus, their manual labeling remains time-consuming and also cost-extensive. In the context of public and general data, a classic solution to limit the cost and time of annotation is to use crowd-sourcing, but this is not possible in domains that require the availability of domain experts to do the annotation such as medical imaging [186]. Moreover, some visual tasks are very demanding because they require a very precise annotation as for example for the semantic segmentation task. At last, the quality of the annotation is also highly dependent on the ability to collect and prepare a dataset, representative of the intended task. Annotation cost is thus a significant issue and major limitation for **supervised learning**, which requires to train the model using fully annotated training data.

  To answer to this strong limitation of supervised models, a natural solution is to develop label-efficient learning schemes that prevent the need of large annotated datasets. In the absence of annotations, **unsupervised learning** [146, 222] is used to train a model that learns the underlying structure of the data distribution. In **semi-supervised learning** [171], a combination of supervised and unsupervised learning is used. The main idea is to use a small amount of labeled (or annotated) data and to leverage a large amount of unlabeled data. **Weakly-supervised learning** [254] alleviates the burden of obtaining high-quality costly or impractical labeled datasets by assuming low quality annotations, i.e. inexact, noisy or incomplete annotations. At the frontier between semi-supervised (small annotated dataset) and weakly supervised learning (incomplete annotation), **active learning** proposes to select a small amount of relevant data which needs to be labelled in order to have the highest impact in the training of a supervised model. The transferability of learned deep representations, mentioned above, has also boosted the development of so-called **transfer learning** [236] and **domain adaptation** [228] approaches. In these paradigms, we assume the availability of a source domain, for which we have a large set of high-quality annotated data to train a model, that is then adapted to a target domain less rich in terms of annotated data. At last, another important recent learning setting to prevent the need of large an-

notated datasets is **self-supervised learning** [107, 146] which is a meet between unsupervised and supervised learning with the automatic building of labels for unsupervised data using some pretext tasks.

These different learning schemes answer to the annotation cost with varying success. Annotation cost of unsupervised algorithms is low but its effectiveness is constrained by the cluster assumption, which assumes that the samples assigned to different clusters are semantically different [42] which is often not satisfied in real world applications. It also fails to capture the dataset semantics with the same degree of refinement and performance as their supervised or semi-supervised counterparts [14]. Annotation cost is still important for semi-supervised learning with the performance depending on the size of labeled dataset [76]. Moreover, the effectiveness of semi-supervised approach also depends on strong assumption on the semantics of data (i.e. cluster, manifold and smoothness assumption) that is not necessarily verified on real world data [223]. Both transfer learning and domain adaptation have been effective to reduce the annotation cost, but are constrained by an assumed *similarity* between the source and the target domain [260]. Finally, the performance of self-supervised learning depends on the ability to design an effective pretext task [111], which can be difficult for high expertise domains.

Thus, we argue that despite the recent advances in exploiting unlabeled data, high quality labeled data used in supervised or semi-supervised learning is crucial. Indeed, most of the advances in computer vision tasks use some form of supervision [53, 148]. In addition, the expertise and knowledge of domain experts can be contributed to the machine learning system during the annotation process [82]. Annotation is also notably more important in computer vision tasks as compared to tasks in natural language processing, due to the well known semantic gap [208].

- **Annotation should be a continuous and dynamic process** Most datasets can be considered as static once curated and deep learning models are generally learnt on stationary batches of training data. Nevertheless, in practical applications, new data might be acquired continuously and thus deep model should account for situations in which information becomes incrementally available over time. It is the case of a large number of real world applications. For instance in data analytics [63] or robotics [51], the model needs to adapt to the changing environment. In cases where it is assumed that one has access to all previously acquired data, the problem becomes trivial, although time and resource intensive, as all acquired data can be used to train the model in a single task. This methodology is very inefficient and also hinders learning new data in real time. When access to the old data is limited or impossible, the problem becomes much more complicated. In particular, deep learning models suffer from **catastrophic**

forgetting [154] where the old information is lost upon re-training to learn new information. If nothing is done to prevent this phenomenon, predictions for past classes become random or nearly so. It is particularly true for deep learning algorithms which heavily rely on labelled data. It thus appeals for annotation or training schemes that take into account the dynamic nature of the targeted domain.

In the literature, various approaches have been proposed to tackle the problem of dynamic domains with different motivations. **Lifelong learning or continual learning** [174] continuously learns on new data while retaining knowledge learned in the past. **Meta learning** methods [134] handle a sequence of tasks, but with the objective to train an efficient learner on new task. Thus, meta learning methods try to extract information during the training of the previous tasks which would make learning the new task easier. This learning strategy has been extensively used in the context of **few shot learning** in which we want to learn with very few samples [231]. Note that meta-learning approaches are different from lifelong learning since they do not have the same emphasis on retaining the previously learned task as the latter [30].

Humans beings and also animals have the ability to continually acquire and expand their knowledge by interacting with ever changing environment [22]. This ability is essential to design models which improve over time without having to learn the model from scratch every time new information is presented [81]. It is an open area of research and an important step towards creating artificial general intelligence [68].

- **Dataset Bias** Another important limitation due to the data-dependence of deep models is their high sensitivity to **dataset bias** [219] which affects most of the real-world applications. Several imperfections such as noisy labels or imbalanced distributions can constitute dataset bias [217]. Dataset bias has recently come into spotlight in vision tasks, mainly due to face recognition applications which show algorithm negative bias towards categories of population which are less represented in training datasets. Authors in [116] provide a systematic literature review of the problem of bias in facial recognition software and highlight the role of training data in instilling bias in the algorithm. This bias is also present in very controlled domain. For instance, in the medical domain, it was shown in [129] that gender bias has a very strong effect in computer-aided medical diagnosis and that it is the main factor of sub-optimal predictions for under-represented gender.

  Class imbalance is a major issue that is mostly neglected when working with academic datasets. These datasets can be considered as optimised for learning since their classes are represented in a balanced way, i.e. the number of instances of each class in the learning dataset is balanced. In practice, it is prudent to consider that data sets are always imperfect.

These imperfections can result from issues in the data acquisition process or from various inherent complexities in real word data. Class imbalance [85] appears when some classes in the dataset are over-represented or under-represented compared to the other classes. The datasets built for real-life applications are often imbalanced and classes of interest specially under represented compared to other classes which are frequent. For instance, in medical imaging, imbalance is encountered between the pathological cases and the normal cases since instance with pathological anomalies can be rare or unique [151]. Learning from imbalanced data, i.e. with minority and majority classes, leads to a prediction bias towards majority classes. This negative effect is well studied for classical machine learning methods as described in the two following surveys [86, 108]. Similar study [23] has been conducted recently on deep learning algorithms with a similar conclusion on the negative effect of imbalance on the prediction performance.

1.3 . Machine learning deployment needs iterative learning schemes

In real-world applications that deploy machine learning models, it is common to assume an iterative scheme where the performance of the model is monitored continuously during the deployment and can be updated with new acquired data as illustrated in Figure 1.4. While this aspect is often considered in the domain named ML ops (Machine Learning Operational) in which a ML life cycle is considered, this iterative scheme is rarely taken into account in research where we only consider the three classical steps of training, validation and testing. In this thesis, we study this iterative scheme in which we assume that new data has to be taken into account iteratively in order to maintain the model performance.

Two scenarios are possible when updating the model with more data :

1. the new data that is acquired or annotated comes from the same domain (i.e. same semantic classes)
2. new data comes from a different domain where either new classes are introduced.

In both cases, considering new data is challenging in the context of deep models in partly due to the data-dependent limiting factors described in the previous section. We thus propose to tackle these two scenarios by adapting two iterative learning schemes to the context of deep learning with incremental, limited and imbalanced data. To tackle the first scenario in which new but limited data of a given domain has to be annotated continuously, we build on the **active learning** scheme. For the second scenario in which new data contains samples from previously unseen classes, we build on **class incremental learning**. We briefly present these two iterative learning scenarios in the following.

1.3.1 . Towards active learning and its limitations

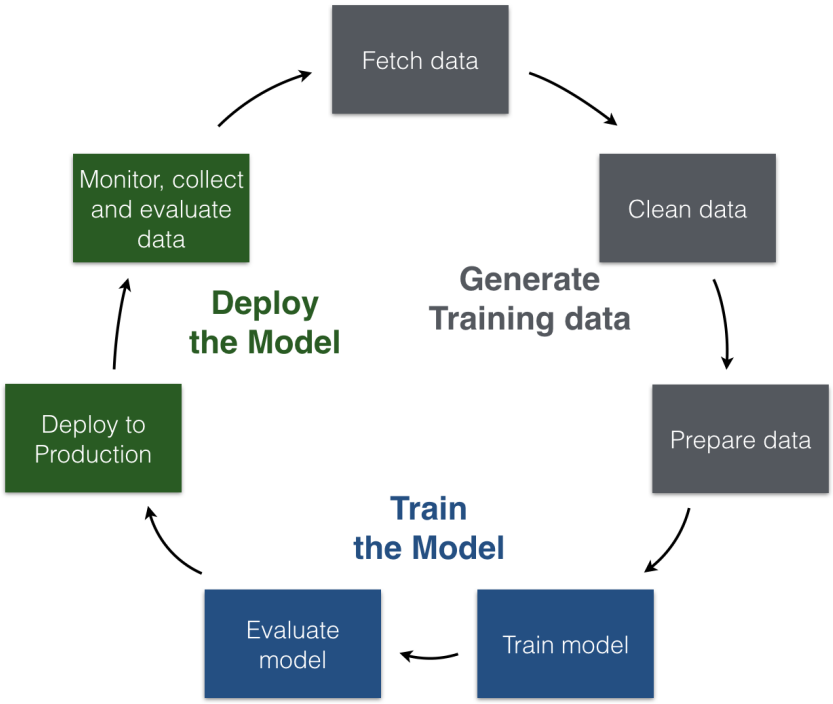


Figure 1.4 – Overview of machine learning pipeline. It shows the iterative scheme commonly used in practical applications. The model's performance is monitored during deployment and additional new data can be used to update the model.

Active Learning (AL) [199] attempts to tackle the problem of annotation cost of large dataset for supervised learning. Under the assumption that not all samples are equally valuable to the model, AL tries to select the most important samples for manual annotation. AL is usually deployed in an iterative fashion. A fixed number of samples is selected per iteration and annotated in order to retrain the model which becomes gradually stronger. The selection strategy can be conducted with different but complementary objectives : maximizing **informativeness** where samples which are likely to bring new information are selected [202, 32, 196, 15] or maximizing **representativeness** where the main criteria is to ensure a diverse set of samples in order to learn a strong representation of the unlabeled dataset [198, 139, 36].

Recently AL has regained interest in the context of deep models. For instance, [189] provides a recent survey on deep active learning (DAL). In this thesis, several aspects have motivated the use of the AL scheme. First, it answers to both the annotation cost and is well-suited to iterative learning scheme deployed in real world applications. Moreover, AL is a human in the loop paradigm and it enables to add continuously high-level expertise in the learning process through annotation. It also brings more explainability [64] since it allows the expert to observe the evolution of

the learning model and its prediction on unlabeled data. As such with the help of human expertise, it might be possible to ascertain which concepts have been learnt by the model and what is remaining [215]. Finally, in AL, the expertise is used to annotate raw data, unlike unsupervised or self-supervised learning setting where annotation is done on clustered data, which could be biased [224]. Nevertheless, despite these recent works on deep active learning, there are still open-issues that limit its use in practical scenarios.

1. **Cold-start Problem :** Active learning learns in an iterative setting. It selects a fixed number of samples at each iteration and then updates the model using the samples annotated to this point. Here, two main problems are encountered : (1) the selection of the initial set of samples to train the first model starting the iterative process and (2) unstable probability estimate of deep learning models when they are trained with limited data. These two issues are referred in the literature as the **cold-start problem** [59, 119, 199, 256]. It requires the labelling of a large enough initial subset to start the AL cycle. This issue is reinforced for deep learning models which are data intensive.

2. **Combining informativeness and representativeness :** Classical acquisition functions either optimise informativeness or representativeness objectives. The joint use of the two objectives can be challenging due to different nature of the two selection strategies [199]. The informativeness measure selects samples which are likely to add missing information to the model, based on the predicted probability distribution. The representativeness measure selects samples in order to well represent the overall input patterns of unlabeled data, based on their representation in the embedding space. The two measures convey complementary information and a few set of approaches have attempted to tackle their combination in order to select samples which are both representative and informative [79, 99, 235]. Works to combine these two criteria could help to select more valuable samples, but joint optimization is still an open issue.

3. **Imbalanced learning in AL :** As said in the previous section, the class imbalance problem is present in most practical datasets. It is argued in [50] that AL can help in class imbalance learning by focusing on samples near the class boundaries where the imbalance is observed to be less than overall distribution. A margin exhaustion criteria can also be used to limit the selection of samples from majority classes [9]. However, it is also shown in [9] that unless processed, high degree of imbalance can have adverse effects on the selection process, with biased model having a preference for selecting samples from majority classes. Active learning for imbalanced dataset adds another component to the selection criteria to ensure that the imbalance in the unlabeled set is not transferred to the selected set.

1.3.2 . Towards class incremental learning and its limitation

Class incremental learning (CIL) [26] aims to add new classes to the learning model, while also retaining the efficiency for the past classes. As discussed earlier, the main challenge in learning with dynamic domains is catastrophic forgetting [154] where models lose the previously learned knowledge, when re-trained with new data. As such, it would be necessary to re-train the models with both previous and new data, leading to higher cost both in terms of computational resources to train on larger datasets as well as memory cost of storing all the past data.

Our motivation to build on CIL is derived from its ability to deal with dynamic domains which might be encountered in real world applications. Further, CIL forms a part of the bigger aim of continual or lifelong learning where new information can be continuously assimilated in the model. The re-usability of the learned information is thus essential to limit the computational cost of learning the new model from scratch with both old and new data. Moreover, CIL also allows to reduce the memory usage by preventing or limiting the amount of past instances required to be stored. The access to old data may be restricted or impossible due to several factors such as : data removal on the Web and in stream data processing [77], privacy in the medical domain [225] or limited resources in embedded systems [175]. Thus, incremental learning is highly desirable in these dynamic domains to boost the re-usability of learned knowledge for efficient resource usage, while also reducing the dependency on past instances.

1. **Catastrophic forgetting :** This is the main challenge in incremental learning, which affects not only deep neural networks but also multi-layer perceptrons or even classical machine learning classifiers. Hence, the problem has been recognized for neural networks as early as 1989 [153], where the researchers found that neural networks lose the past knowledge when trained for a new task. As the model is updated with new data, the learned weights are overridden by the new task and thus degrade the performance on older tasks. Thus, the re-training of the model without data from previously learnt classes leads to the catastrophic forgetting of previously acquired knowledge. A stability-plasticity dilemma [156] is presented for designing model that learn continually over time. Lifelong learning systems have to be stable enough to retrain the old information while at the same time showing plasticity to acquire new information without catastrophic interference with the already acquired information. A large number of works [43, 242, 245] take inspiration from the human and animal brain to tackle the stability-plasticity dilemma. Despite some advances, deep learning models are very far away from displaying similar capacity of lifelong learning as humans and animals and preventing catastrophic forgetting in deep incremental learning models is still an open issue.
2. **Model Calibration :** The authors of [73] show that deep neural nets are

over-confident on their predictions, with models often giving wrong predictions with a very high probability score. A model is said to be calibrated if the average softmax probability of the predicted class for all samples in a dataset corresponds to the accuracy of the model over that dataset. Well-calibrated models have confidence levels aligned to the model accuracy and thus give valuable information of how likely the model is to be correct or incorrect. In several scenarios, it can be important to take into account the confidence of the model on the predictions. This is also the case when decision has to be taken based on prediction of more than one model. In incremental learning, calibration is important as well-represented classes of new task show higher level of confidence as compared to classes for old task which are bounded by fixed memory [114]. The incremental model gets further mis-calibrated over times as the number of classes increases and each old class is then represented with fewer samples.

3. **Imbalanced learning in IL :** Dealing with imbalance is an integral part of incremental learning where access to old data is limited, leading to imbalance between the old and new classes. If the initial dataset is balanced, as it is assumed in existing incremental learning [25, 109, 188], the associated imbalance profile is binary, with new classes having a large number of images and past classes having a small but identical number of images. If the dataset itself is imbalanced, as it is often the case in real contexts, the inherent imbalance is added to the incremental one and the resulting imbalance profile can be more complex. Thus, techniques have to devised to not only tackle the imbalance between old and new classes, but also between the new classes.

1.4 . Contributions

In this thesis, our objective is to provide new methodological tools to answer to several limitations that affect the deployment of deep neural models in real word applications. These limitations are threefold : the need of large annotated data to build efficient and domain-knowledge aware models, the need of iterative learning schemes in order to take into account the dynamic behavior of a majority of real word applications and the imbalanced nature of most real world datasets. Based on the two iterative learning schemes presented in the previous section, the active learning scheme and the class incremental learning ones, we apply and evaluate our solutions on visual recognition tasks such as image classification but our solutions are generic and could be apply with small adaptation to other visual tasks such as image segmentation or object detection as well as in tasks containing textual data or one dimensional time-series data. In both settings, we consider the presence of dataset imbalance as a core issue and propose solutions to mitigate its affects. An overview of the issues tackled in the different chapters along with the corresponding

contributions is given below and shown in Figure 1.5.

- Our first contribution, described in **Chapter 3**, proposes a new approach, named single stage active learning, to answer to the cold start problem in deep active learning. As said before, the iterative active learning process needs an initial labeled dataset, large enough to be able to be used to kick start the iterative learning process. Taking inspiration from transfer learning and domain adaptation, we propose to use a general purpose representation that is learned on a source domain. This proposition is in line with other efficient label learning schemes, and in particular few-shot learning scheme in which it has been shown that state of the art results can be obtained by a good learned representation [216]. The principle of our approach is to use a representation learned on a large labeled **source dataset to represent samples** and to select, according to their representations, a diverse set of samples to present for annotation. Our approach also assumes that the unlabelled dataset can be imbalanced and our approach can limit this imbalance.
- We then proposed to improve the the classical iterative active learning setting that assume, contrary to the previous contribution, a sufficient initial labeled subset of the target domain to answer to two of its limitations. In **Chapter 4**, we focus on a better management of the imbalance of the dataset. We propose a new selection strategy that **prioritizes minority classes** for balanced and informative selection. We also compare the methods from the first contribution which rely on a source domain with the methods developed in the iterative setting.
- In **Chapter 5**, we propose a new strategy to combine the objectives of informativeness and representativeness in the selection. We introduce a novel acquisition function which selects samples based on estimates from models learned in current and previous iterations. Samples for which there is a maximum shift towards uncertainty between the last two learned models predictions are favored. The choice is made to select samples for which the model is most likely to forget and thus find difficult to learn.
- Our last contribution deals with the case where new classes can be seen in the production data (i.e. dynamic domain) and build on incremental learning. Our work focuses on limiting the catastrophic forgetting while taking care of the class imbalance. We propose a detailed study of imbalanced incremental learning with focus on calibration methods whose objective is to reduce the prediction bias between majority and minority classes. We also propose two novel calibration methods and compare their performance to the existing ones. This part of our work is presented in **Chapter 6**.

We also propose a state of the art on active learning and incremental learning, along with the challenges related to class imbalance problem in Chapter 2. The

manuscript is ended by a general conclusion and some perspectives in Chapter 7. Our works have been published in recent international conferences and journals.

- **Chapter 3** Aggarwal Umang, Adrian Popescu, and Céline Hudelot. "Active learning for imbalanced datasets." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2020.
- **Chapter 4** Aggarwal Umang, Adrian Popescu, and Céline Hudelot. "Minority Class Oriented Active Learning for Imbalanced Datasets." 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021.
- **Chapter 5** Aggarwal Umang, Adrian Popescu, and Céline Hudelot. "Optimizing Active Learning for Low Annotation Budgets" arxiv 2201.07200 in cs.CV .
- **Chapter 6** Aggarwal Umang, Adrian Popescu, Eden Belouadah, and Celine Hudelot. "A comparative study of calibration methods for imbalanced class incremental learning." Multimedia Tools and Applications (2021) : 1-20.

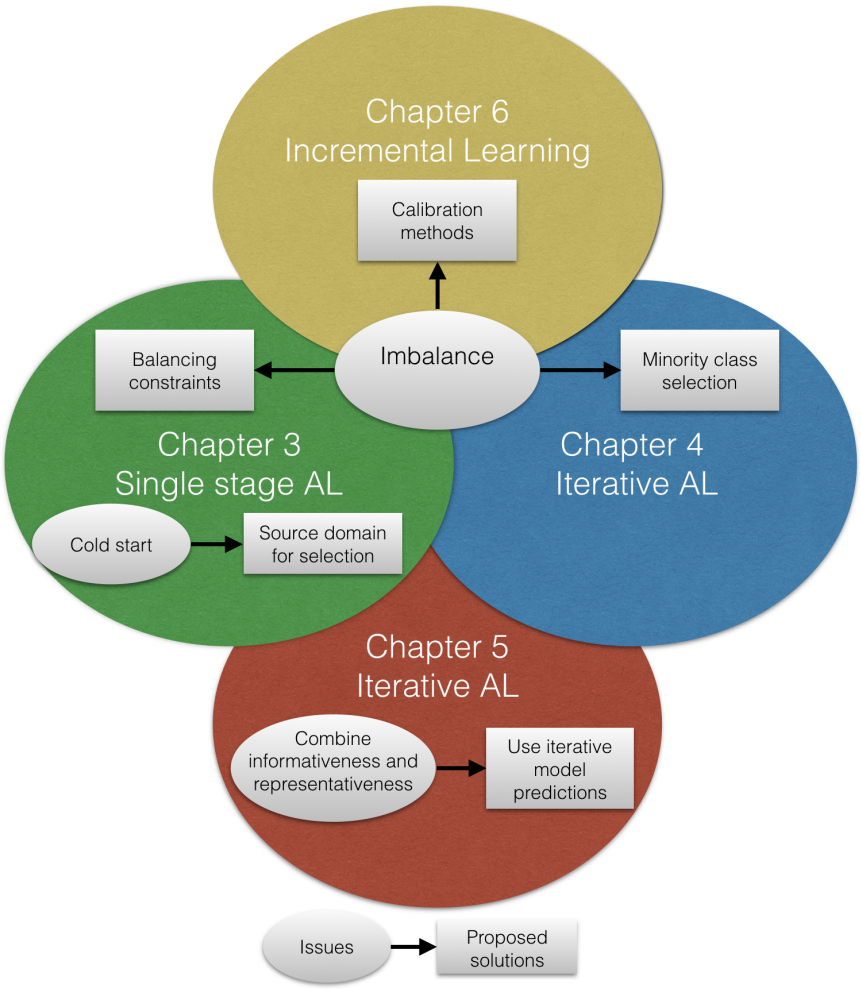


Figure 1.5 – An overview of the contributions in the different chapters. Chapter 3, 4 and 5 tackle the problem of annotation cost in the context of active learning. Chapter 3 proposes a single stage active learning setting in order to answer to the cold start problem in AL by using the predictions from a model learned on a large annotated source dataset. Chapter 4 devises approaches to manage imbalance in iterative AL setting by prioritizing selection of samples from minority classes. Chapter 5 explores the use of active selection based on iterative probability estimates. This selection strategy allows to design a method to combine the informative and representative objectives in AL. In Chapter 6, the problem of catastrophic forgetting in dynamic domains is tackled in the context of incremental learning. In this chapter, in addition to imbalance between old and new classes present in incremental learning due to bounded memory, we tackle the problem of dataset imbalance. We also explore a fine-tuning scheme along with post-calibration methods instead of more complicated distillation losses [188] and propose methods to improve the model calibration.

2 - Related works on active and incremental learning

We start by discussing the techniques used to deal with class imbalance which is a major component of dataset bias in real-world datasets in Section 2.1. We then describe various approaches to deal with highly data intensive nature of deep learning in Section 2.2. Further, we discuss the related work in the two learning schemes, active learning (Section 2.3) and incremental learning (Section 2.4) studied in our work.

2.1 . Imbalanced Learning

Imbalanced learning is a learning setting in which classes have different prior probabilities in the distribution [85]. Learning with imbalance is one of the main challenge when working with datasets from real applications which often present some kind of imbalance, with one or more classes being under- or over-represented in the distribution. Learning from such datasets leads to a stronger feature extractor for samples in the **majority classes** as compared to the ones in the **minority classes**. This is a natural consequence of having more samples in majority classes as compared to minority classes. This further leads to learn classifiers that are biased to predict the majority class [86, 108, 23].

Two kinds of minority exists : **absolute rarity** and **relative rarity** [85]. Absolutely rare classes do not have enough examples to learn a generalized representation for the class, whereas for relatively rare classes, classes may have enough examples to learn a good generalized representation, but performance over these classes is hampered as they exist in presence of majority classes. The problem of learning a generalized representation for absolutely rare classes is named as **rare class problem** [234]. Absolute and relative rare classes can co-exist leading to the combination of rare class and class imbalance problem.

In addition to imbalance **between** classes of the dataset, we can also have **within** class imbalance [85]. Within-class imbalance corresponds to the case where a class is composed of a number of different sub-clusters and these sub-clusters do not contain the same number of examples. This might lead to the classifier learning only on the dominant sub-cluster and not the complete representation for a class, resulting in mis-classification for images if they do not belong to the dominant sub-cluster.

Imbalance is a serious problem in many real world applications where identification of minority classes is of real interest. For example, in case of fraud detecting in banking system, only a few samples of fraud will be available compared to non-fraud samples. Other application domains suffering from imbalance are listed in

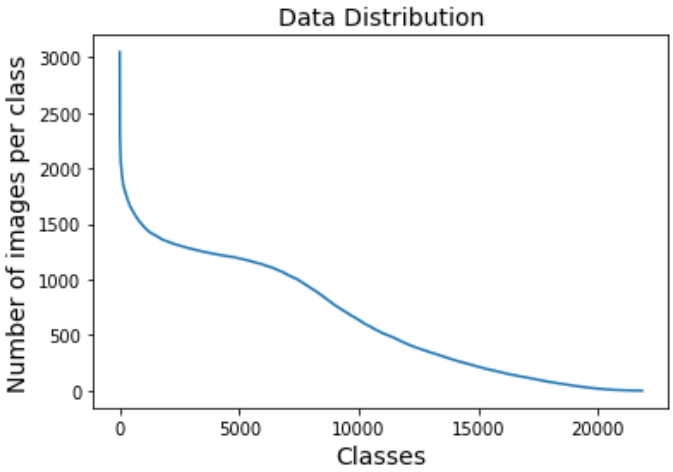


Figure 2.1 – Image distribution per class for full set of ImageNet leaf classes. The data distribution over full set of ImageNet leaf classes is highly imbalanced with number of samples in class varying from more than 3000 to less than 150 samples. The data distribution over ImageNet leaf classes has a mean of 592 and a standard deviation of 508. It is in contrast to ILSVRC dataset which is a balanced subset of ImageNet containing 1000 classes with a mean of 1231 images per class and standard deviation of 70.

[121] with bio-medical applications [184, 135, 103] and social data mining [240] being some of the important ones.

A wide majority of deep learning works, often implicitly, make the hypothesis that training datasets are balanced or nearly so. It is a consequence of benchmarking deep learning advances with academic datasets built for research purpose. This is, for instance, the case of the ImageNet $LSVRC$ effort [193], that contains 1000 leaf classes, well represented in the dataset and thus balanced. The $ILSVRC$ training set used in [188] has a mean of 1231 images per class, with a standard deviation of 70. However, an analysis of the full set of ImageNet leaf classes [38] shows that image counts per class are highly variable, with a mean of 592 and a standard deviation of 508.

Several approaches have been proposed in the literature to deal with imbalanced datasets. A comprehensive review of the methods is covered in [86, 108, 121, 113]. Proposed methods can be divided into two categories [86] : data-Sampling methods and algorithm based methods. Hybrid approaches combining data-sampling and classifier-level methods are also studied in [147]. We discuss the most important ones below.

2.1.1 . Data-sampling methods

Data-sampling methods mitigate the bias towards majority classes by balancing the training dataset. Balancing can be achieved either by randomly undersampling the majority classes or by randomly oversampling minority classes [75]. The main risk of undersampling is that it can lead to incomplete representation of classes, in particular in case of within-class imbalance. Informed undersampling [126] partially solves this problem by avoiding to select images which are close to class boundary and thus informative for the discriminative problem. Oversampling methods are sensitive to overfitting and affect the generalization ability for the minority classes. Oversampling can also increase the training time significantly in case of high level of imbalance on large datasets.

Synthetic Minority Oversampling Technique (SMOTE) [29] that consists in generating samples for minority classes by interpolating between existing minority samples is an influential solution to this problem. SMOTE has been extended and improved in [12, 78, 150], by modifications in the interpolation method. It basically creates synthetic samples for a dataset by applying simple arithmetic transformations to actual samples. SMOTE is related to data augmentation, but acts only on minority classes whereas data augmentation is more global.

A two phase learning approach [133, 84] which combines undersampling with transfer learning is also studied in literature where the model is first pre-trained with undersampled data and then fine-tuned with complete imbalanced data. These works show an improvement in the performance of minority classes but are tested only on highly specific domains, plankton classification in [133] and brain tumor data in [84].

The direct comparison of these techniques is not straight-forward due to different datasets, imbalance levels and models used. [92] performed experiments with different imbalanced versions of Cifar10 and concluded that random oversampling can be effective in addressing slight class imbalance. [23] provides the most comprehensive comparative study of these methods in the context of deep learning for different imbalance levels and datasets. While they showed some improvements for Cifar10 and Mnist [132], oversampling, undersampling and oversampling were shown to have detrimental effect at higher levels for imbalance of the $ImageNet$ dataset [23]. Further, [23] did not find any improvement of using two phase learning over random undersampling in one phase.

2.1.2 . Algorithm-based methods

Learning from imbalanced dataset leads to classifier biased towards the majority classes. Hence, another line of work deals with imbalance at classifier level. Cost sensitive methods assign higher cost of misclassification to minority classes, forcing the classifier to adapt to the imbalanced dataset [49]. An initial method of estimating cost by moving threshold at the inference step is presented in [255]. Some recent methods [101] [117] integrate the cost in the loss function of deep

learning models.

Some works [229, 141] also propose new loss functions to give more weights to samples from minority classes. In [229], as an alternate ive to mean square loss, authors propose to compute the total loss as a mean of all individual class losses, so that all classes contribute equally irrespective of the number of training samples in each class. Further, loss of each class is computed as a mean of false positive and false negative to better capture the errors in minority classes. Focal loss [141] deals with extreme imbalance by modifying the cross entropy loss to reduce the impact of samples that are easily classified. The easily classified samples are most likely to come from majority classes and thus it reduces the effect of majority classes on the loss function.

Thresholding works as a post-scaling step and modifies the decision threshold of the classifier to counter the bias toward majority classes. A formulation of thresholding is proposed in [190] where the outputs are modified using the prior class probabilities. This allows to post-scale the output of the classifier according to the number of training samples in each class. While very simple, thresholding outperformed a large array of data-sampling and classifier-level methods for object recognition in the context of deep learning models as shown in [23]. It is thus a very competitive method in dealing with imbalance and is evaluated in the context of both active learning and incremental learning in our work.

2.2 . Data-efficient deep learning approaches

As discussed in the introduction, the availability of large and perfectly labeled datasets is a central requirement to train robust deep learning models. These large labeled datasets can not be built easily for different reasons :

1. data acquisition - costly or unfeasible acquisition of data due to expensive hardware or privacy concerns in some cases like medical data.
2. data annotation - time required to annotate large amount of data and also the level of expertise required for annotations in some domains [186].

We present in the following different learning paradigms than the supervised one that prevent the need of large annotated datasets.

2.2.1 . Data augmentation

Data augmentation makes the assumption that more information can be extracted from the dataset by using augmentations on training samples [205]. It has been traditionally used to improve the generalization ability of the model towards new instances. Various augmentation techniques have been introduced in the literature. *Data wrapping techniques* are transformations that preserve the label of the augmented sample. They include geometric transformations such as random rotation, cropping, flipping etc., color space augmentations [214] or kernel filters [115]. Generated samples are added to the training data for supervised lear-

ning. Several techniques have been devised for deep learning models such as feature space augmentation [41, 144], adversarial training [70, 138], generative adversarial networks [69], neural style transfer [60] and meta-learning schemes such as Auto Augment [31] which generates samples similar to the ones in the training dataset. Although data augmentation has been shown to improve the generalization ability of the model, it does not comprehensively solve the annotation cost issue of build large labeled datasets for supervised setting. [104] also shows that data augmentation with low labeled data can lead to model uncertainty. A detailed survey on data augmentation techniques is provided in [205].

2.2.2 . Transfer learning/ Domain Adaptation

Works on transfer learning [236] and domain adaptation [228] have been proposed to transfer the knowledge from a source domain with large annotated data to a target domain with limited data. A common strategy, known as fine tuning, entails two steps. First, a model is pre-trained on a well and large annotated source domain (e.g. ILSVRC). Then, the model is partially re-trained on the target domain using a limited number of annotated data in the target domain. Pre-trained models can also be used to provide powerful feature extractor, which can be universally applied across different tasks. Unsupervised domain adaptation is a particular transfer learning approach in which we assume that only unlabeled samples are available for the target domain. Transfer learning is based on several assumptions such as the availability of a large annotated dataset from a source domain and some similarity relations between the source and the target domains. While transfer learning can reduce the need of annotated data from the target task, it does not completely solve the problem of annotation cost.

2.2.3 . Semi-supervised learning

Semi-supervised learning can be used to limit the annotation cost by using unlabeled data, that it easier to collect, in order to alleviate the lack of labeled data. Several semi-supervised methods, as detailed in [171], have been proposed in the literature and can be broadly categorized as consistency based [18], proxy labeling [192], generative models [168, 118] and graph-based methods [210, 33]. The current state of the art methods [209] use a consistency loss [18] to train the model to give consistent output on unlabeled samples despite small perturbations using data augmentation or dropout parameter. Semi-supervised learning is part of weakly-supervised learning (WSL) [254, 250] which encompasses scenarios of inexact [80], noisy [226] or incomplete supervision.

Semi-supervised learning has performances that are close to supervised learning, with considerably less annotations for some datasets [209], but it remains some limitations. Semi-supervised learning imposes some constraints such as cluster and manifold assumption on the data [171] which might not be fulfilled for more complex datasets such as ones containing overlapping classes. Most semi-supervised learning works assume the presence of annotated validation set for tuning the hy-

perparameters, which can be larger that the labeled set [169]. Such an assumption is unrealistic in real-world scenarios where large validation sets are not readily available. Finally, annotation cost is also important for semi-supervised learning with the performance depending on the size of the labeled dataset [76].

2.2.4 . Unsupervised learning

Unsupervised learning uses only unlabelled data to learn the underlying structure of the distribution [146, 222]. Even though recent advances have been made in unsupervised learning [96], performances are normally lower than in the supervised setting. The unsupervised setting assumes a clustering hypothesis, where samples belonging to the same cluster are assumed to be semantically similar. It is not always the case, in particular for fine classes or complex classes for which the cluster assumption might not hold. In order to make the clusters semantically meaningful, a labelling or a fine study of the clusters, involving experts can also be needed [224]. Expert knowledge is therefore involved at the end of the learning pipeline. Recent guidelines [160] recommend the use of expert knowledge on raw data rather than on machine learning outputs which could be biased. Further, over or under-clustering could also make the annotation process difficult for the expert.

2.2.5 . Self-supervised learning

Self-supervised learning involves automatic building of labels using some pretext tasks to learn representations that can then be used for clustering, segmentation or classification tasks. In general, the pretext task involves withholding a part of the data, and train the model to predict the missing data. A large number of pretext tasks have been proposed in literature such as patch prediction [44, 162], adding color to images [130, 251], using adversarial training [45, 48], predicting rotations [65], or discriminating instances [239, 90, 158]. The performance of self-supervised learning depends on the ability to design an effective pretext task [111]. The representations learned using the pretext task are then tested for the target classification task by fine-tuning using annotated data [224]. Consequently the expert knowledge is integrated at an intermediate step of the learning pipeline. While self-supervised learning can aid in learning an effective initial representation, the annotation cost of samples for the final task still remains an important concern. An end-to-end pipeline is also studied in self-supervised learning where representation learning and cluster association is done simultaneously using a contrastive loss [107, 146]. This is similar to the unsupervised learning pipeline, except the use of pretext task and hence it also suffers from concerns of annotating on biased model outputs for classification task.

Techniques / Scenarios	Supervised learning	Data augmentation	Semi-supervised Learning	Transfer/ domain Adaptation	Self-supervised	Active learning
High data acquisition cost	-	+	-	+	--	+
High annotation cost	--	+	+	+	+	++
Availability of related source domain	~	~	~	++	~	~
Imbalance	~	+	~	~	~	++
Human in the loop	-	-	-	-	-	+

Figure 2.2 – Various learning schemes/techniques that deal with limited or no labeled data can be deployed based on their applicability in different contexts. + and - signs shows that positive or negative impact of the learning schemes in tacking the various scenarios. In the context where the data acquisition cost is high, the applicability of techniques dependent on large amount of data such as supervised learning (SL) and semi-supervised or self-supervised learning is affected. Techniques such as transfer learning and domain adaptation can mitigate the need of large dataset depending on the availability of a related source domain. Active learning (AL) can significantly reduce the amount of annotated data by selecting only the most relevant samples for annotation. While imbalance adversely affects most of the learning setting, data augmentation can potentially deal with some level of imbalance by augmenting the samples from minority classes. AL has also been shown to tackle the problem of imbalance by selecting samples near the classifier boundary where the imbalance is lower. Finally, recent emphasis has been laid to include the human in the loop of machine learning pipeline. AL is most suited due to its iterative nature and has also shown to increase the explainability of models.

2.3 . Active Learning

Active Learning (AL) is a sub-field of machine learning that consists in selecting carefully instances to annotate in order to enable the model to learn on these few highly important instances [199]. The objective of AL is thus to reduce the annotation cost while achieving the required model accuracy. Active learning is not a new domain and has been extensively used in classical machine learning. AL has been applied mostly for binary classification or one dimensional data [137, 122] using classifiers such as Support Vector Machines (SVMs) [218] or K-Nearest Neighbours (KNNs) [106]. Most recently, AL has been applied to deep learning models for high dimensional problems such as image recognition [198, 7, 15], object detection [83, 191] or text classification [5, 197]. These works have shown the promising potential of AL to tackle the data greedy nature of deep neural networks. In Figure 2.2, we position AL according to the other data-efficient learning scheme presented in the previous section.

The iterative setting is commonly used in active learning as shown in Figure 2.3. A majority of AL works [199] assume a weakly supervised setting, i.e. access to a small manually labeled subset which includes all classes of the target domain is provided at the start of the process. This assumption is necessary to kick start the AL procedure by first training a model on the labeled subset. Then, the learned model's outputs are used to select a given budget of data from the unlabeled dataset and submit them to an oracle for annotation. Then, the model can be updated in an iterative manner when new samples are labeled.

In AL, the different approaches can be classified according to the selection strategy [199] as shown in Figure 2.4. These approaches can primarily be divided as :

1. **Selective sampling** : stream-based sampling or pool-based sampling.
2. **Query synthesis**

The first way to perform AL is **selective sampling**, where the samples are selected from the given input distribution. Two setting are possible in selective sampling, depending on the availability of unlabeled samples. In **stream-based setting**, the learner decides to select or discard a sample from an incoming stream. This setting is effective when the underlying distribution changes over time. The decision to select the sample is generally based on a threshold on a predefined criteria (eg. uncertainty). This could be sub-optimal when the data distribution is static.

The second approach, named **pool-based setting**, examines all the unlabeled samples to select the subset to annotate. Compared to stream-based setting, query decisions are not taken individually but using evaluation and ranking on the set of unlabeled data. This approach is commonly used for deep active learning [15, 198]. Indeed, while in classical AL, samples are generally selected one at a time for annotation, it is not feasible in deep learning due to its computationally expensive

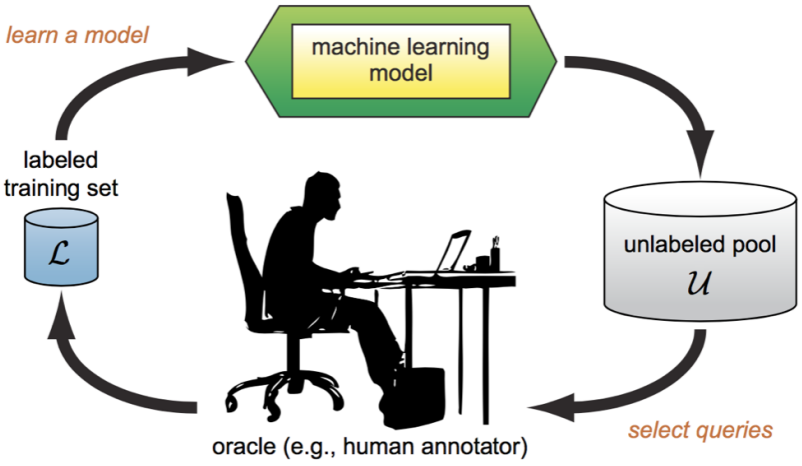


Figure 2.3 – Classical active learning cycle, Source : [199]. A labeled initial subset is used to learn a predictive model. The learnt model is then used to provide predictions on the unlabeled dataset. A subset of samples defined by the AL budget is selected to be annotated by the oracle from the unlabeled dataset. Selection is done in "select queries" phase based on the model predictions on unlabeled dataset and an AL objective which is defined by acquisition function. The selected subset is annotated and added to previously labeled dataset. The model is then re-trained using the larger dataset in the next cycle.

training and also the fact that addition of single sample would not bring a significant change in the model.

An alternative to selective sampling is to use **membership query synthesis**, where the model generates a synthetic sample which is then annotated by a human expert. This setting has been studied for classical machine learning but it has limited applicability for deep models. Indeed, deep neural networks are highly non-linear and might generate samples that are incomprehensible to human annotator [127].

In the following, we first present active learning techniques developed with classical machine learning models in section 2.3.1, followed with works which have been developed for deep models in section 2.3.2.

In the following, we first discuss the active learning techniques globally in Section 2.3.1, and then study the specific issues and their related works when applying active learning with deep neural networks in section 2.3.2.

2.3.1 . AL works

The literature on AL approaches is quite large and an exhaustive review is provided in [108, 199]. More recently, [159, 189] provide a review of active learning

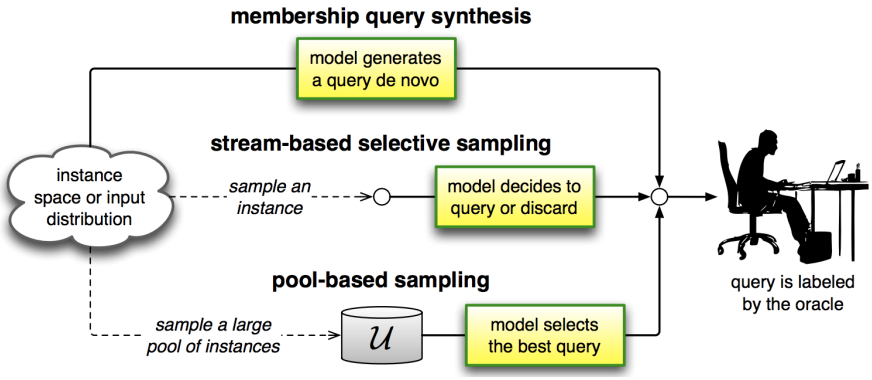


Figure 2.4 – Different query strategies that can be used in AL pipeline to process the unlabelled dataset, Source : [199].

in deep learning context.

Here, we propose to formally define the problem of active learning in order to set up the notations for the rest of the manuscript. We introduce the notion of a target domain $\mathcal{D}_{\mathcal{T}} = (\mathcal{X}, \mathcal{Y}_{\mathcal{T}}, p_T)$ as the combination of an input space $\mathcal{X}$, an output space $\mathcal{Y}_{\mathcal{T}}$ and an associated probability distribution p_T. In the rest of the thesis, we will be interested by a C class classification problem that means that we consider that $\mathcal{Y}_{\mathcal{T}}$ is a label space $\mathcal{Y} = \{1, ..., C\}$. A domain is represented by two datasets sampled over the space $\mathcal{X} \times \mathcal{Y}_{\mathcal{T}}$: a large unlabelled dataset of n_T samples, $\mathbb{D}_T^U : x_i \in \mathcal{X}$ for $i = 1..n_T$ and a labelled dataset $\mathbb{D}_T^L : (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}_{\mathcal{T}}$. This latter can be initially empty (hard cold start setting of active learning).

We also consider a model $\mathcal{M}_{\mathcal{T}}$ of parameters θ_T and a loss function $l(., .; \theta_T)$: $\mathcal{X} \times \mathcal{Y}_{\mathcal{T}} \to \mathbb{R}$ parameterized over the hypothesis class (θ_T). We assume that the model generates predictions in the form of probability distributions $p_{\theta_T}(\cdot|x)$ on $\mathcal{Y}$ for $x \in \mathcal{X}$. Also feature embedding $f_{\theta_T}(x)$ can be extracted from the model. The objective of active learning is to build, most of the time iteratively, the set $\mathcal{D}_{\text{train}}$, used to learn the model $\mathcal{M}$ in a supervised way, i.e. by minimizing the expected loss on it. $\mathcal{D}_{\text{train}}$ is initialized with the initial labelled dataset $\mathbb{D}_T^L$ and the objective of active learning is to carefully choose most relevant data point from $\mathbb{D}_T^U$, ask its annotation by an oracle and add to the current $\mathcal{D}_{\text{train}}$. In the following, we also note $a(\{x_i\}, \theta|\mathcal{D}_{\text{train}})$ an acquisition function which scores unlabelled data points $\{x_i\}$ (singleton (stream based) or a set of samples (pool-based)) using the current model parameters learned on the current $\mathcal{D}_{\text{train}}$. Most of the time, the acquisition function is a way to implement a query strategy.

Using this notation, we can simply defined the active learning with the following optimisation problems to select the next data point :

$$\underset{x^* \in \mathbb{D}_T^U}{\arg\min} \mathbb{E}_{x,y \sim p_T}[l(\mathbf{x}, \mathbf{y}; \theta|\mathcal{D}_{\text{train}})] \tag{2.1}$$

, where

$$x^* = \underset{x \in \mathbb{D}_{\mathbb{T}}^{\mathbb{U}}}{\arg\min} a(\{x_1, ...x_b\}, \theta | \mathcal{D}_{\text{train}}) \tag{2.2}$$

Below, we present only the most important query strategy methods used in AL and we classify them according to the criteria used for the selection.

1. **Informativeness** A first class of methods is based on the informativeness of samples. These methods try to estimate the new information that would be added to the model by annotating a given sample and adding it in the training dataset. We present below the main approaches that are based on the estimation of class membership probabilities and are thus dependent on the model parameters learned on the current $\mathcal{D}_{\text{train}}$. The posterior probability distribution is used and evaluated for determining whether an unlabeled data sample should be queried for label or not.

   - **Uncertainty-based measures** In these approaches, the informativeness criteria measures the ability of the instances in reducing the uncertainty of the model. These approaches favor least certain samples that are often at the borders of classes but while they can help in improving the decision boundary of the classifier, they might not be most representative for the data distribution as a whole [199]. We present in the following the three main approaches used to estimate uncertainty.

     Entropy sampling is based on the Shannon information view on uncertainty [202]. It measures the difference between all predictions as defined in information theory. According to our notations, if we consider that the current model generates predictions in the form of probability distributions $p_{\theta_T}(\cdot|x)$ on $\mathcal{Y} = \{1, ..., C\}$ for $x \in X$ and for the current model parameter θ_T then, the uncertainty of a sample x is computed according to the following equation :

$$H(x, \theta_T) = -\sum_{c=0}^{C} p_{\theta_T}(y_c|x) * \log p_{\theta_T}(y_c|x) \tag{2.3}$$

     Margin Sampling estimates the uncertainty with the margin in the confidence of top two predictions. It thus prioritizes samples for which the margin is the lower. It is shown to be effective in AL task as it is equivalent to select samples based on distance from the decision boundary.

$$marg(x, \theta_T) = p_{\theta_T}(\hat{y_1}|x) - p_{\theta_T}(\hat{y_2}|x) \tag{2.4}$$

     with : $\hat{y_1}, \hat{y_2}$, the top-2 predicted classes for test sample x.

Least confidence sampling prioritises the samples which predict a class with least confidence. It thereby selects samples whose maximum predicted probability has minimum value.

$$lc(x, \theta_T) = p_{\theta_T}(\hat{y}|x) \tag{2.5}$$

where : $\hat{y}$ is the predicted class for sample x.

- **Query by committee** Another way to ascertain the informativeness of samples is to take into account the probability estimates from multiple classifiers. Query by committee [203] selects samples that have the maximum disagreement among predictions from different classifiers. The ability to create classifiers which are consistent with the annotated dataset but contain disagreements is, thus, a central requirement to find informative samples. Various methods to create such classifiers have been proposed in literature, for instance with different random parameter initialization [203, 11], ensemble-based boosting or bagging [1, 54] or with different regions of feature space for different classifiers [163].
- **Other methods** An approach to define informativeness of a sample estimates the change in model's prediction, if the label of the sample is known. [201] estimate the model change for gradient-based learning models by computing the gradient of the loss function for each sample. This method can become computationally expensive for larger models and with large unlabelled dataset. Expected error reduction approaches [259, 161] define informativeness by estimating the ability of sample to reduce the generalisation error. The model is re-trained for each query and the expected error of the query is approximated over all the possible class labels. This makes the approach highly computationally expensive and has only been studied with classical models for binary classification problem.

However, informativeness based methods on batch do not take into account the diversity of samples in a batch. It could lead to selection of high informative but similar samples which is often detrimental to model performance.

2. **Representativeness** A second class of methods was proposed to improve the representativeness of the selected samples. The objective is thus to select samples which represent the underlying distribution of the dataset.

   Farthest-first traversal [94], that selects the most distant sample from the last selected sample, has also been tested to maximize representativeness. More recently [62] implement farthest-first traversal with deep models for selection over long-tail distribution.

Diversity based approaches have also been studied to improve the representativeness of selected samples. In [56], discussion on various similarity measures (i.e. cosine, gaussian) is conducted to ascertain similarity between samples and to enable the selection of the most diverse ones. Clustering-based techniques [105] select the most representative samples from different clusters to annotate diverse, high density samples. Recently in deep learning, discriminative active learning [67] selects diverse samples by solving the binary classification problem of discriminating between labeled and unlabeled samples in the representation space. The selection is done with the motivation to make it difficult to distinguish between labeled and unlabeled samples.

Information density is a simple strategy which consists in selecting samples from high density regions to avoid outliers. Information theory methods are studied in [152] for active selection for text classification task. [237] selects the high density samples by prioritising sample with minimum average distance to all other samples.

Coreset [198] is a recent method that solves the K-center problem as shown in Figure 2.5. It tries to minimize the distance between any unlabeled point to its closest labeled point. Hence at every step, it selects the point which is at a maximum distance from its closest labeled point to cover the representation space with least number of points.

$$core(\mathbb{D}_T^U, \mathbb{D}_T^L) = \max_{\forall x_u \in \mathbb{D}_T^U} \min_{x_l \in \mathbb{D}_t^L} d(f(\theta_T, x_u), f(\theta_T, x_l)) \tag{2.6}$$

where $core(\mathbb{D}_T^U, \mathbb{D}_T^L)$ returns a sample from unlabeled dataset $\mathbb{D}_T^U$ using the labeled dataset $\mathbb{D}_T^L$, $d(f(\theta_T, x_u), f(\theta_T, x_l))$ is the distance between a labeled point x_l from $\mathbb{D}_T^L$ and an unlabeled point x_u from $\mathbb{D}_T^U$ and $f(\theta_T, x)$ is the feature embedding of sample x from the model $\mathcal{M}_\mathcal{T}$ with parameters θ_T.

These methods require to compute the distance matrix of all the samples in the unlabelled dataset, which is costly and is hence limiting for large datasets.

For batch-based setting, both informative and representative methods can be sub-optimal since they optimise for only one of the criteria.

3. **Hybrid approaches** A set of approaches have tackled the combination of the two selection criteria in order to select samples which are both representative and uncertain. Early efforts propose to exploit **clustering algorithms** to combine the two objectives. [35] uses hierarchical clustering followed by pruning of non-informative samples, while [20] implements an approximation of spectral clustering followed by a selection of samples from the cluster boundary. [177] applies a k-means clustering on a set of the most uncertain samples and selects the most representative ones. More

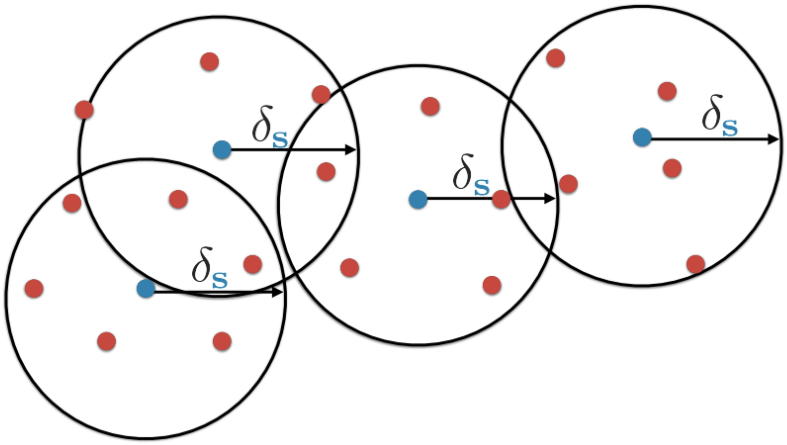


Figure 2.5 – Coreset [198] is used to select the unlabeled sample which is at a maximum distance to its closest labeled sample. This can be visualized by drawing circles of same radius δ_S from each labeled sample (represented in blue) such that they cover all the unlabeled samples (represented in red). The size of circle would thus depend on the sample which is furthest away from its closest labeled data point. The objective is to provide coverage over all the unlabelled samples, while minimizing the radius of these circles. Thus the unlabelled sample at the circle boundary and not represented in any other circle is selected to ultimately reduce the radius of circle.

recently, [252] optimises k-means clustering by including the informative weights of samples in the optimisation algorithm. Similarity of samples implies the computation of the distances between all samples and hence can be computationally expensive.

An **information density-weighted** approach is presented in [200]. It weights the informativeness of a sample with its average similarity with other samples.

QUIRE algorithm [102], shown in Figure 2.6, takes a min-max view of active learning by using the prediction uncertainty of the samples based on labeled and unlabelled data to quantify informativeness and representativeness respectively. In [27], entropy and KL-divergence are combined to obtain uncertain and representative examples. Batch Active learning by Diverse Gradient Embeddings **(Badge)** [7] is a recent work which samples from a hallucinated gradient space using K-means++ clustering. The gradient embedding of a sample has both a magnitude and direction vector associated to it. The batch of samples having high gradient magnitude with diverse gradient direction is selected to meet the two objectives.

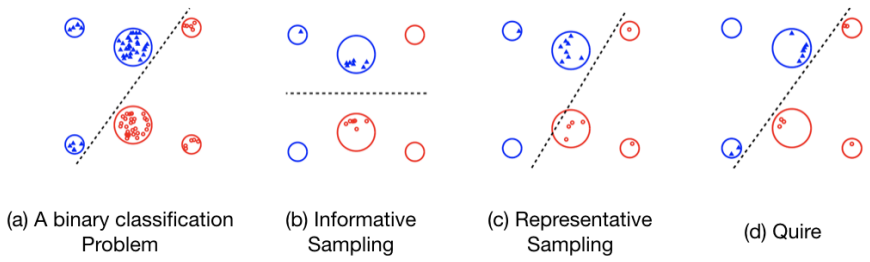


Figure 2.6 – Quire [102] method shows the advantage of selecting both informative and representative samples. (a) shows all the unlabeled samples and the true decision boundary for the binary classification (b), (c) and (d) show the samples selected with informative, representative and combined (QUIRE) objective.

2.3.2 . Deep AL- some limitations and solutions

Deep learning has given great results with automated feature extraction from a large amount of high-dimensional data. AL has the potential to expand the application of deep learning models by reducing the annotation cost. Although a large number of acquisition functions have been defined for classical AL, the adaptation of these methods to deep learning models is not straight-forward. The major issues in DAL arise due to data intensive nature of deep models, their mis-calibration and the joint learning pipeline to combine the AL with deep models. We briefly describe these limitations in the following as well as approaches to narrow them.

1. **Data-intensive deep models** Data intensive nature of deep learning also requires both larger amount of initial data and larger batch size in subsequent iterations of AL cycle.

   An early effort to add additional data in deep active learning is presented in Cost-Effective Active Learning (CEAL) [227]. As shown in Figure 2.7, CEAL increases the amount of labelled data by assigning pseudo labels to high certainty samples based on model prediction and using the oracle to assign true labels to most uncertain samples.

   Further, several ways to use complementary data through techniques such as data augmentation, semi-supervised learning etc. have been studied in the literature. We present the most relevant works from each technique in the following.

   - **Data augmentation using GANs** Another way to increase the labelled data is data augmentation using Generative Adversarial Networks (GANs) as shown in Figure 2.8. For example, GAAL [257] uses generator network in AL to generate samples which might have additional information. However, GANs are likely to generate high quality

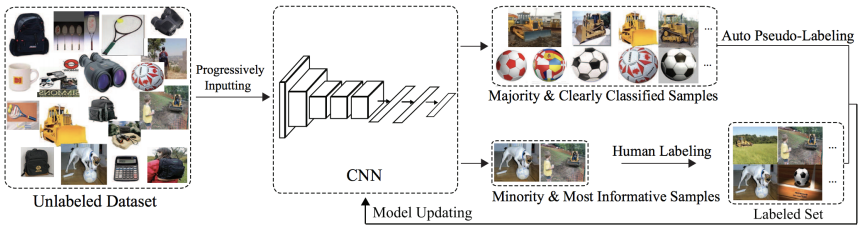
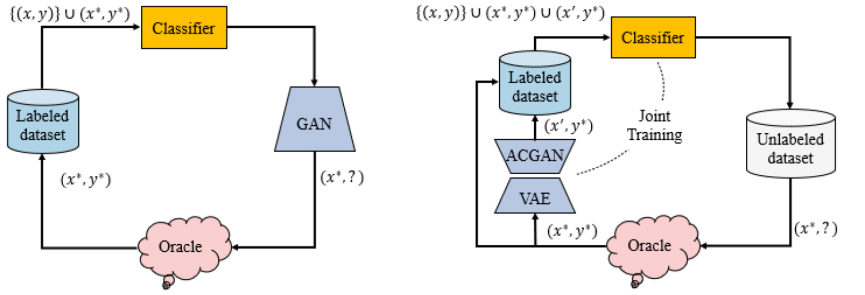


Figure 2.7 – CEAL [227] adds the most confident samples with their pseudo labels for training and gives the least certain samples for annotation to get the true label before adding them to the training dataset.

samples, similar to those that have been learned correctly, and hence have no guarantee of having more information than in the labeled dataset. Bayesian generative active deep learning (BGADL) [220] tackles this problem by proposing a joint training of generative and classification models and uses bayesian acquisition function, such as BALD [98], to select the informative samples from labeled dataset. The generative model is then used to generate samples which are similar to already selected samples to ensure that generated samples have high informative value. An implementation of GANs for augmenting samples of minority class for credit card fraud detection is presented in [52].

- **Semi-supervised learning** Several works add the semi-supervised objective to active learning by using the unlabelled data to train the model. An approach to find the optimal acquisition function using feature density matching between unlabelled dataset and weakly supervised validation data is presented in [71]. Inspired by MixMatch [18], consistency based semi-supervised learning [59] has been used for AL task by selecting samples which give inconsistent predictions for different permutations as shown in Figure 2.9. Semi-supervised learning is difficult to generalize, particularly at lower annotation budgets. It also comes with an added complexity in terms of implementation and computational resources. More importantly, most of the methods in this category use an additional labeled validation set to optimize the parameters. It is not realistic to assume that such a set exists at the beginning of the AL process.

- **Adversarial learning** has been used recently in AL [207, 247] to use both labelled and unlabeled data as well as training using adversarial samples. Variational autoencoder is used in [207] to learn a latent space by using an adversarial network to discriminate between labeled and unlabelled data points. [247] expands the above approach by adding a certainty indicator in the discriminator to ascertain the importance



(a) Generative adversarial active learning (GAAL).

(b) Bayesian generative active deep learning (BGADL).

Figure 2.8 – Generative adversarial active learning [257] uses data augmentation through the generator model (GAN). The augmented sample x^* is given to the oracle to annotate and added to the labeled dataset. Bayesian generative active deep learning (BGADL) [220], a joint training of classifier and Variational Autoencoder Generative Adversarial Networks (VA-GAN) allows to generate augmented sample $(x^{'}, y^*)$ for a labeled sample (x^*, y^*), both of which are directly added to the training dataset.

of sample. [48] uses the smallest adversarial attack to ascertain the distance to the decision boundary for CNN models. This is shown to be an effective alternative to margin based approach on SVM classifier which inherently selects samples close to the decision boundary. The importance of sample is ascertained by finding the closest adversarial sample by using several small random variations of the samples. This can be computationally limiting for large datasets.

2. **Cold-start problem** The data intensive nature of deep learning exacerbates the cold start problem of active learning, with larger amount of annotated instances required to learn a model which provides reliable estimates suited for AL task. The initial set is selected randomly in most works [15, 198] on AL to kick start the iterative cycle. The estimation of amount of samples required to learn an efficient initial model for AL task can be tricky. It is shown that selecting a too small or too large set of initial samples can lead to sub-optimal performance for a given total annotation budget [59].

   Very few works tackle the problem of active selection with no initial selected subset to train the first model. Active incremental fine-tuning (AIFT) [256] proposes to use iterative fine tuning in order to improve AL for bio-medical images. The main advantage is that the labeled seed samples are no longer needed. Instead, a network learned on an external and independent dataset is used for fine-tuning. Image patches are used to calculate entropy and diversity over image regions and thus to select relevant examples. In [243]

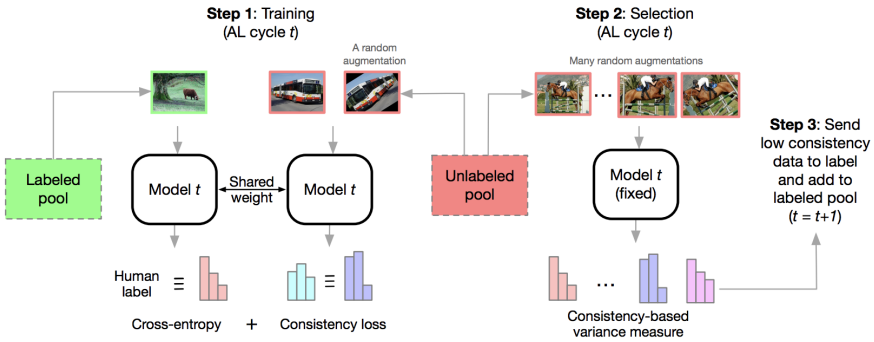


Figure 2.9 – Semi-supervised learning in AL [59] exploits both labeled and unlabeled data during model training with cross-entropy loss for labeled samples and consistency loss for unlabeled samples. Further during the selection phase, unlabeled sample with higher consistency loss on its various augmented instances is selected for annotation.

tackles the cold-start problem for language modelling by using loss functions of Bert model [40] to select the most efficient initial samples for labelling.

3. **Miscalibration of deep model predictions** Another issue in DAL is the problem of mis-calibration in the deep learning models. Some studies [74] have shown that deep models show higher level of confidence on their predictions than the accuracy of the predictions. In the context of DAL, mis-calibration makes the probability estimates unreliable for AL task.

   To tackle this issue, a bayesian perspective is taken to have a more realistic evaluation of uncertainty measures. Bayesian probabilities were introduced as better estimate of uncertainty by combining probabilities of several runs of the model [57]. Monte Carlo (MC) dropout exploits the softmax predictions of a deep model with random dropout masks to generate to model uncertainty. Ensemble models have also been used to acquire multiple probability estimates. In [15], an ensemble of model snapshots is created by using a cyclic learning rate. This design choice is important in order to limit the computational effort needed to create the ensemble. Coupled with a variation ratio function [112], ensembles are shown to outperform MC dropout. MC-dropout and ensembles increase the computational complexity of the AL process since multiple inferences are needed for each image.

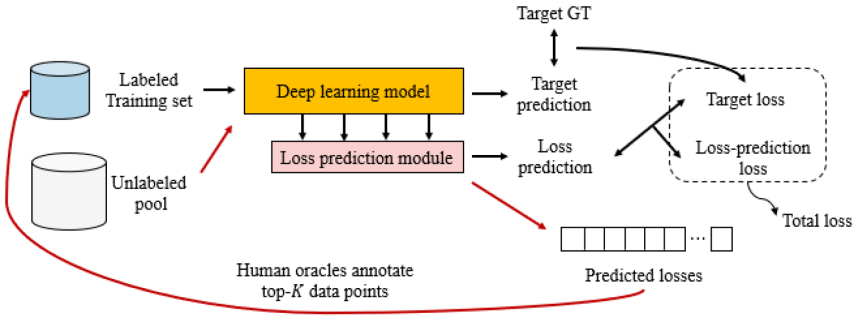


Figure 2.10 – Learning loss for active learning [241]. During the training phase, represented with black arrows, labeled samples are used to learn the parameters of both the model and the loss prediction module by optimising for both target loss and loss prediction loss. During the query phase, represented with red arrow, loss prediction module is used to select the samples with maximum predicted loss.

4. **Pipeline inconsistency between AL and DL**

   Classical AL involves training only the classifier over extracted feature representations, while deep models jointly optimises both feature representation and the classifier. As such, simply fine-tuning the deep model in AL context might lead to some divergence issues.

   To tackle the pipeline inconsistency between AL and DL, some works propose a combined framework that meets the AL and DL objectives. For example [241] (see Figure 2.10), introduces a parallel parametric loss prediction module in the DL pipeline which learns to predict losses on unlabelled data at the model training time. The module is then used to predict unlabeled samples which might have not been correctly learned. Active learning with partial feedback [100], tackles the multi-class classification problem by asking yes/no questions to annotator. Labels are then progressively pruned as the process advances to minimize the manual labeling effort. In contrast to classical AL, Deep AL learns both feature extractor and classifier during training. [91] exploits the predictions from different layers of the feature extractor along with classifier predictions to extract better uncertainty estimates as shown in Figure 2.11.

   Some works have used concepts from **meta learning** and **reinforcement learning** to optimize the selection strategy using previous knowledge. Meta learning approaches have been tried in AL to select samples from previous selection strategies. In [119], sample selection strategies are learned from an ensemble of multiple previous AL problems. A major drawback of meta-active learning is considerably larger amount of data needed to train the meta-learner. A reinforcement learning objective is used in [173] to derive

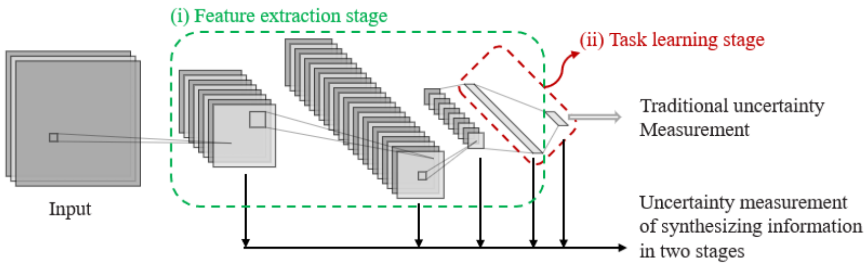


Figure 2.11 – Towards better uncertainty sampling : Active learning with multiple views for deep convolutional neural network [91] estimates the uncertainty using prediction from different layers of the feature extractor.

an active learning strategy directly from data. An approach to learn the data representation along with the selection heuristics is presented in [10]. While very interesting, approaches from reinforcement learning are not entirely suited for large-scale image classification task.

2.3.3 . Imbalance in AL

Active learning has been shown to be useful in mitigating the transfer of imbalance from unlabeled to labeled datasets in the context of classical machine learning algorithms. The authors of [50] concluded that samples close to the decision boundaries are likely to have less imbalance than the overall distribution and this observation is thus used to drive sample balancing. Uncertain samples are selected in [50, 258] along with a margin exhaustion criteria to limit the selection for majority classes. As a result, the selection process stops when all samples close to the margin are exhausted so as to avoid selecting samples for majority classes. However, the authors of [9] show that a high degree of imbalance has an adverse effect on the selection process. The resulting model will be biased towards selecting samples from majority classes. Also, the prioritization of samples which are close to the hyperplane can have negative effects, as it fails to create a good representation of the minority class. Particularly, the problem can be made worse when the minority class contains several concepts or is not easily separable from a majority class [87]. Consequently, a diverse representation of minority classes should be targeted. Cost-sensitive SVM (CS-SVM) was exploited as an effective way to handle a skewed data distribution during active learning [50, 258]. Other works try to explicitly favor the selection of minority class samples to reduce imbalance. The mis-classification component is used to penalize selection of majority classes with boosted SVMs in [261]. In [87], the authors propose to prioritize samples which are nearest neighbours of minority classes. Note that most of these works are designed for binary classification with shallow classifiers and are not easily transferable to the multi-class classification problem using deep models tackled in our work.

Some works tackle the problem of imbalance in the context of deep active learning. Certainty-Based Active Learning (CBAL) [55] algorithm uses local behaviors in specific areas to identify query samples. It determines the query probability of samples in relation to all samples within the neighbourhood. Similarity active learning (SAL) [248] actively learns a similarity model to recommend unlabeled minority class samples for manual labeling, while high confidence unlabeled samples for majority classes are automatically pseudo-labeled. [19] proposes a method which optimises batch selection to have minimum confidence and minimum redundancy in the selected set. Any of the uncertainty measure can be used to ascertain confidence while cosine similarity between samples is used to measure redundancy. Further, samples with similarity to minority classes are prioritized. Active learning important sampling (ALIS) [230] proposes an AL framework, which consists of selecting important majority-class instances and generating informative minority-class instances. Nevertheless, their approach considers large annotation budget which is unrealistic.

2.4 . Incremental Learning

Incremental learning is a machine learning method where a model is trained repeatedly to extend its knowledge with new incoming data. As discussed earlier, the main challenge in incremental learning is catastrophic forgetting [154], i.e. the tendency to forget previously learned information when new data is incorporated. It occurs whenever access to old data is constrained or impossible. Catastrophic forgetting is a major issue in the context of deep learning models which jointly optimise feature extraction and classification [17]. Several recent surveys [174, 13, 179, 37] have been done to summarize the issues and solutions for implementing incremental learning in the context of deep learning models. We briefly present them following their classification into three main categories.

2.4.1 . Parameter isolation methods

The first class of methods focuses on changing the neural net architectures to incorporate new knowledge. Influential works include *Growing a Brain* [233], progressive neural networks [195] or lifelong learning with a network of experts [4]. These methods are interesting but their complexity grows when new classes are added incrementally. Notably, inference time will become longer as the model grows and the scalability of these methods is consequently reduced.

2.4.2 . Regularization-based methods

This class of methods imposes regularization constraints to try to preserve the past information. The past information can be stored using the model parameters or classification outputs for the past classes. Learning-without-Forgetting (LwF) is an influential method in this class and is presented in [140]. The algorithm does not rely on past data and exploits knowledge distillation [93] to reduce the discrepancy between the outputs of old classes with past and new model. A warm-up step which freezes old parameters and trains the new ones is first performed in Lwf. Then a joint training is run until convergence.

Alternatively, some works try to estimate the importance of model parameters for different tasks. While learning a new task, changes to important parameters are penalized to mitigate forgetting. Variational Continual Learning (VCL) [164] uses an approximation of bayesian inference to prevent critical model parameters from changing drastically with new data. [3] builts on VCL to lower the memory cost for determining regularization strengths, while [246] explores a more realistic task agnostic setting.

2.4.3 . Rehearsal based methods

This group of methods is based on training the model with new data along with limited number of samples for old classes. These methods have constant model complexity, except for the classification layer which integrates new classes, and are more fitted for large scale content analysis. Most of them require a bounded

memory in order to partially avoid catastrophic forgetting. The memory related constraint is more acceptable than model complexity growth when analyzing large datasets since the inference time is not influenced by the use of memory.

iCaRL [188] builds on top of LwF in that it combines classification and distillation losses for each incremental state of the algorithm. A first important difference with LwF is that a bounded memory is allowed to store exemplars of old classes. As more classes are added, the number of images per old class is reduced to fulfill the memory constraint. Class exemplars are selected using a *herding mechanism* which gives priority to images that are closest to the class mean. A second difference is related to the classification mechanism. Instead of using the class activations of the deep models, a nearest-mean-of-exemplars is implemented. The $iCaRL$ average top-5 accuracy on $ILSVRC$ is 62.5%. An $iCaRL$ analysis [109] indicates that the most important algorithm components are the bounded memory and the distillation loss. The herding mechanism and the nearest-mean-of-exemplars classification seem to matter less. Recently an end-to-end incremental learning scheme with a bounded memory was introduced in [25]. The main modification compared to $iCaRL$ resides in the proposal of a loss function which includes separate distillation terms for each incremental batch. In addition, data augmentation and balanced fine tuning are used to reduce the effect of data imbalance between old and new classes. Top-5 accuracy on $ILSVRC$ is 69.4%, to be compared with 62.5% obtained by $iCaRL$. Interestingly, the use of herding to store exemplars is only marginally useful (0.5 points) compared to random selection. This finding confirms the $iCaRL$ analysis conclusions from [109].

BiC [238] tackles the bias against old classes, by adding a linear layer with two learnable parameters after the classification layer. A small part of the dataset is reserved to learn the parameters of the bias correction layer. The training is done in two steps. The model and classifier weights are learned first, then the bias correction layer is learned using only the reserved dataset. In $LUCIR$ [97], authors proposed three balancing constraints at the time of training to mitigate the imbalance bias between old and new classes. Firstly, they modify the distillation loss component using cosine normalization to counter larger weights and biases for new classes. Further, they exploit the observation that imbalance is less pronounced at the classifier margin to introduce a margin loss function which is less susceptible to imbalance. Finally, a less forget constraint is introduced which complements the distillation loss by encouraging the orientation of features extracted by current network to be similar to those by the original model.

An alternate approach [204, 8] to store the samples from old classes is using GANs to learn the data distribution and then using synthetic samples generated by GANs for rehearsal. Though interesting, it adds the computational complexity of training GANs continuously.

In chapter 6, we study the problem of class incremental learning with a bounded memory and model size. The focus is thus on rehearsal based methods where

exemplars from each class is stored for future training.

2.5 . A brief summary and our positioning

Table 2.1 presents a summary of various issues in implementing deep active learning along with possible solutions proposed in the literature. We make some contributions to tackle these different problems in the following chapters. In Chapter 3, we propose a single stage selection strategy which uses a larger labeled source dataset to select a diverse and balanced initial labeled set, as an alternative to random sampling which is used in most works [15, 198]. DAL is quite data intensive and several works such as data augmentation using GANs [257, 220, 98], semi-supervised learning [71, 59], adversarial learning [207, 247, 48] have been proposed to add complementary data in the pipeline. In our work, we test learning shallow classifier over strong pre-trained representations as an alternative to more data intensive classical fine-tuning methods used in all these works. The proposed training scheme is particularly interesting as it solves the issue of unreliable uncertainty estimates of deep models at lower budgets, while also mitigates the pipeline inconsistency problem of DAL, since only shallow classifiers are learned as in classical AL setting. In batch pool AL, a combination of informative and representative objective is desired. In chapter 5, we propose a novel way of selecting samples by using predictions from successive iterative AL models which allows us to select informative and diverse samples. Imbalance is a central part of our work and we propose diversification and balancing constraints to select a balanced set of most representative samples for both single stage and classical iterative AL setting in Chapter 3 and 4 respectively.

In the incremental learning setting, several approaches have been studied to elevate the problem of catastrophic forgetting. We focus on rehearsal based methods, which assumes the possibility of storing some samples for old classes. In chapter 6, we study a set of calibration methods to tackle the bias between old and new classes.

Issues in AL	Possible solutions
Cold start Problem	Random selection of initial labeled seed set [15, 198] AIFT [256] uses pre-trained model for selection **Designing single stage AL setting (Chapter 3)**
Using complementary data	Data augmentation using GANs [257, 220, 98], CEAL [227] Semi-supervised learning [71, 59] Adversarial learning [207, 247, 48] **Using pre-trained models(Chapter 3,4,5)**
Unstable uncertainty estimates	Bayesian [57], Ensemble [15] **Using shallow classifiers over fixed representations(Chapter 3,4,5)**
Pipeline inconsistency	LAAL [241] Using hidden layers [91] Meta learning [119] / Reinforcement learning [173] **Using shallow classifiers over fixed representations(Chapter 3,4,5)**
Hybrid Selection to combine informative and representative objective	Clustering-based methods [35, 20, 177, 252] Quire [102], Badge [7] **Using iterative model predictions(Chapter 5)**
Imbalance datasets	Cost-sensitive SVM classifiers [50, 258] Margin-based AL [87] **Designing diversification and balancing constraint(Chapter 3,4)**
Issues in IL	Possible solutions
Catastrophic forgetting	Parameter isolation [233, 195, 4] Regularization-based methods [140, 93, 246] Rehearsal based methods [188, 238, 97] **Study of calibration methods for rehearsal based methods (Chapter 6)**

Table 2.1 – Summary of different issues and the possible solutions in deep active and incremental learning . Our solutions to the different issues are highlighted in bold.

3 - Single stage active learning for imbalanced dataset

Most works in active learning assume the presence of an initial labeled set of samples to start the AL process [15, 198]. We propose to tackle a more realistic hard setting in which we assume the following constraints : (1) we have no knowledge on the target dataset, in the sense that we consider that no initial seed set of annotated samples of the target domain is available (hard cold start problem) and we are even not ascertained of the number of target classes ; (2) we have only a limited budget for data annotation by an expert ; (3) we assume possible imbalance in the unlabeled target dataset.

Our objective, here, is to select a diverse set of samples which best represents all the classes from the unlabeled target dataset, while also ensuring that the imbalance present in the unlabeled dataset is not transferred to the selected set. With these objectives in mind, we propose a new approach that takes benefits from both deep transfer learning and active learning acquisition functions. A pre-trained model, learned on a source domain is used to provide robust estimates of target sample uncertainties for selection, while also providing strong data representation for training the target model on the annotated samples by the oracle. Our main contribution is the adaptation of classical acquisition functions (AFs) to this single stage AL scenario where source and target domains are introduced. We introduce a diversification procedure which selects samples that are predicted as different source class by the pre-trained model. Further, a sample balancing step is introduced which reduces the propagation of imbalance from the unlabeled to the labeled dataset.

The evaluation of the contributions is done with imbalanced versions of four public datasets designed for different visual tasks. Three AL labeling budgets are tested for each dataset. The modified acquisition functions are compared to random selection, to their original formulation and to core [198], a recent geometric-based approach before and after balancing. We take inspiration from works in transfer learning [185, 120] to propose two different schemes of training using shallow classifiers on deep features and fine-tuning the deep learning model. Results indicate that both the modified acquisition functions and sample balancing are useful for three of the four imbalanced datasets. We also provide an analysis on the transferability between source and target domains to legitimate the usability of source domains for the proposed scheme.

The outline of the chapter is as follows. In Section 3.1, we motivate our approach. Then, in Section 3.2, we propose a formalization of our new learning setting along with the definitions of the adapted acquisition functions in subsection 3.2.1. The proposed methods, introducing the balancing and diversification constraints, are presented in Section 3.3. An extensive experimental validation is proposed in

Section 3.4 and we finally derive the conclusions in Section 3.5.

3.1 . Motivations

As mentioned in Chapter 2, classical and deep active learning approaches suffer from the cold start problem. Indeed, a first seed of labeled samples is necessary to initialize the AL process. The size of this initial labeled dataset can be important, in particular for deep learning models [189]. It is an important drawback of the classical scenario leading to selection of samples based on features, which might be weak or unstable [199]. This problem is particularly stringent for deep models which are data-intensive involving the need of larger batch sizes during both initial and subsequent training steps.

In our work, we assume that such an initial labeled dataset is not available for the target domain but that we can exploit robust features learned from a related source domain. The proposed scenario is single stage in nature as the total number of samples allowed by the AL budget is first selected and the AL model is learned on the whole selected and labeled dataset. Moreover, an iterative training of deep models to include each new labeled sample or batch of samples is very expensive and time consuming. The existence of a pre-trained model is also a realistic hypothesis which is extensively exploited in transfer learning [185, 120] and that we propose to exploit in this context.

This chapter combines ideas from active learning, domain adaptation and transfer learning to tackle the cold-start problem. Our objective is to select the minimal subset of samples from a completely unlabeled dataset on a target domain in order to present them to an oracle for annotation by leveraging knowledge coming from the source domain. This is very close to domain adaptation, but our goal is to select a diverse subset which best represents the target dataset, while reducing the annotation cost. Further, we apply transfer learning to use the model that is learned on source domain for training the model in the target domain. Techniques from semi-supervised learning can be easily assimilated to further improve the learned model by also using the rest of the non-annotated data in the target dataset. Nevertheless, semi-supervised learning does not always lead to improvement and can also be detrimental in some cases [181].

AIFT [256] is the existing work which is closest to ours since we also make the assumption that a pretrained model exists and can be exploited to remove the need for a labeled seed set. However, important differences arise from : (1) our focus on imbalance and (2) the criterion used to select candidate samples.

3.2 . Problem Formalization for single stage AL

In this section, we formally describe the problem of single stage active learning. An overview of its pipeline is provided in Figure 3.1.

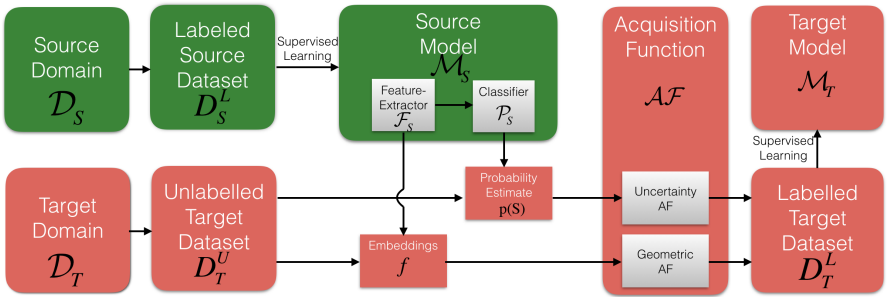


Figure 3.1 – Overview of the single stage active learning pipeline. The source model $\mathcal{M}_S$ trained over the large labeled dataset $\mathbb{D}_S^L$ from the source domain $\mathcal{D}_S$. $\mathcal{M}_S$ consists of the feature extractor $\mathcal{F}_S$ and the classifier $\mathcal{P}_S$. The unlabeled target dataset $\mathbb{D}_T^U$ from target domain $\mathcal{D}_T$ is passed through $\mathcal{F}_S$ and $\mathcal{P}_S$ to extract the features f and predictions p_S for target samples. The AF is applied on these estimates to select the samples to be annotated. The target model $\mathcal{M}_T$ is trained once the $\mathbb{D}_T^L$ is completely filled with b samples.

We consider a source domain $\mathcal{D}_S$ represented by $\mathbb{D}_S^L$ a labeled dataset with $x_i, y_i \in \mathcal{X} \times \mathcal{Y}_S$ for $i = 1..n_S$, i.i.d. realizations of random variables $\mathcal{X}, \mathcal{Y}_S \sim \mathbb{P}_S$ where $\mathbb{P}_S$ is the source domain data distribution, $\mathcal{X}$ is the instance space (in our case the image data), $\mathcal{Y}_S$ is the set of N_S class labels $\{y_1, ..., y_{N_S}\}$ of the source domain and n_S the number of annotated instances.

The target domain $\mathcal{D}_T$ is only represented by an unlabeled dataset $\mathbb{D}_T^U$ in the hard setting here. The objective of single stage Active Learning is to select the best subset $\mathbb{D}_T^L$ from $\mathbb{D}_T^U$ of cardinal b (the budget) for manual labeling in order to maximize the performance of its associated model over the test dataset in target domain $\mathcal{D}_T$. We also consider that $\mathbb{D}_T^L$ is imbalanced, i.e. target classes can be under or over represented. The level of imbalance can be defined, for instance, by using a combination of mean (μ) and standard deviation (σ) of the number of images per known class. The higher the ratio between σ and μ is, the stronger is the imbalance of the dataset.

Our AL scenario encompasses the following steps. First, a deep model $\mathcal{M}_S$ represented by parameters θ_S is learned over the source domain and includes two main components. The first is a feature extractor $\mathcal{F}_S : x \to \mathbb{R}^d$, with d the size of the feature vector. The second is a classifier followed by a soft-max function $\mathcal{P}_S : \mathbb{R}^d \to P_S$ which outputs the probability distributions over the N_S classes of the source domain. The two components of the model $\mathcal{M}_S$ are used to extract features f and predictions p_S for all samples x from $\mathcal{D}_T^U$.

Second, a manually labeled dataset $\mathbb{D}_T^L$ is obtained via the application of an acquisition function $\mathcal{AF}$ [199]. Here we note the two main challenges. As the model used to extract features and probabilities of the target dataset is trained on

the source domain, classical uncertainty-based $\mathcal{AF}$ might be sub-optimal due to dataset shifts [183]. Also the target dataset contains imbalance which gets propagated to $\mathbb{D}_T^L$. Minority classes are likely to be underrepresented or not represented at all, especially for low labeling budgets. We thus introduce :

1. adaptations of uncertainty-based $\mathcal{AF}$ by diversifying samples based on source class predictions
2. a two step acquisition process which first uses $\mathcal{AF}$ to discover classes and then focuses on balancing the number of samples per class.

At last, a model $\mathcal{M}_T$ is trained over the resulting $\mathbb{D}_T^L$. This model can be built either by transferring representations from the initial model or by fine-tuning it. The usefulness of each of the two approaches is determined by the AL budget b and the transferability of features between $\mathbb{D}_S^L$ and $\mathbb{D}_T^L$. We perform cross-validation on the training set to determine which of the options is better in each configuration. Optionally, semi-supervised learning can be then applied to expand $\mathbb{D}_T^L$ into a larger subset $\mathbb{D}_T^S$ but this part of the process is not in focus here.

We now describe acquisition functions defined for the single stage setting.

3.2.1 . Acquisition Functions for single stage AL

In the single stage AL scenario, no manual annotation of the target dataset is available at the start of the process to train a model on the target domain. Thus the uncertainty and representative measures of the completely unlabeled target dataset are computed using the outputs of the source model $\mathcal{M}_S$. First, we define the acquisition functions to give the estimates for samples from completely unlabeled target dataset based on predictions of the source model. These estimates might not be directly meaningful for our task of creating a diverse and balanced initial labeled set and modifications of classical AF are required to include these objectives. In the next section 3.3, we detail our proposed modifications to instill diversification and balancing constraints.

Uncertainty-based Functions

As discussed in Chapter 2, uncertainty-based methods allow an AL method to query the instances on which the model is most uncertain. In single stage AL context, these methods exploit the classifier $\mathcal{P}_S$ obtained with the pretrained model $\mathcal{M}_S$.

Entropy Sampling is based on the global shape of class predictions p_S and is defined in single stage scenario as :

$$H(x, \theta_S) = -\sum_{c=0}^{N_S} p_S(y_c|x) * \log p_S(y_c|x) \tag{3.1}$$

with : N_S the number of source classes.

We consider $ent_{\text{sort}}(\mathbb{D}_T^U)$ as a permutation of the set $\mathbb{D}_T^U$ by ordering its element by *decreasing* value of entropy $H(x, \theta_S)$. D_T^L is obtained by annotating the first b samples from list. This baseline is noted as ent.

Margin Sampling computes the uncertainty of an instance x by comparing its top 2 predictions of the source model. It is defined in single stage scenario as :

$$marg(x, \theta_S) = p_S(\hat{y_1}|x) - p_S(\hat{y_2}|x) \tag{3.2}$$

with : $\hat{y_1}$ and $\hat{y_2}$ are the top 2 predicted classes with $\mathcal{M}_S$ of parameters θ_S for the sample x.

We consider $marg_{\text{invsort}}(\mathbb{D}_T^U)$ as a permutation of the set $\mathbb{D}_T^U$ by ordering its element by *increasing* value of margin $marg(x, \theta_S)$. The baseline is noted as ms.

Least Confidence Sampling selects instances for which the model $\mathcal{M}_S$ gives prediction with lowest probability. It is defined in single stage scenario as :

$$least(x, \theta_S) = p_S(\hat{y_1}|x) \tag{3.3}$$

with : $\hat{y_1}$ the top 1 predicted class for the sample x with $\mathcal{M}_S$.

We consider $least_{\text{invsort}}(\mathbb{D}_T^U)$ as a permutation of the set $\mathbb{D}_T^U$ by ordering its element by *increasing* value of margin $least(x, \theta_S)$. $\mathbb{D}_T^L$ is obtained by annotating the first b samples from list. The baseline is noted as lc.

Geometric-based Functions

Geometric approaches are based on building a subset which best represents the complete dataset using the feature extractor $\mathcal{F}_\mathcal{S}$ of the pretrained model $\mathcal{M}_S$ on the unlabeled dataset.

Coreset

As discussed earlier, coreset selects unlabeled samples which are at a maximum distance from their closest labeled samples. We implement this method in single stage setting by randomly selecting the first labeled point and then solving Equation 3.4 :

$$core(\mathbb{D}_T^U, \mathbb{D}_T^L) = \max_{\forall x_u \in \mathcal{D}_T^U} \min_{x_l \in \mathcal{D}_T^L} d(f(x_u), f(x_l)) \tag{3.4}$$

with $d(f(x_u), f(x_l)$ the distance between the labeled point x_l from labeled set $\mathbb{D}_T^L$ and unlabeled point x_u from unlabeled set $\mathbb{D}_T^U$. The selected sample is then moved from unlabeled set $\mathbb{D}_T^U$ to labeled set $\mathbb{D}_T^L$. The process is continued till the budget is exhausted. The baseline is noted as $core$.

3.3 . Proposed method

3.3.1 . Diversified Certainty-based Functions

We propose a diversification strategy based on source model's top predictions on target dataset. Our aim is to select a diverse set of samples which covers the

maximum number of classes from unlabeled target dataset. To meet this objective, the proposed diversification strategy selects samples which have different top source classes as prediction. The hypothesis is that samples which are predicted as different source classes are likely to be different from each other. This strategy operates under the assumption that, due to representation transferability, a mapping between classes in the source and target domains occurs. Even if imperfect by nature, class mapping might help to partially counter the effects of imbalance and to discover a broader range of classes compared to random sampling.

The uncertainty acquisition functions introduced in the last section act as a base for our proposed selection strategy. We perform an inversion of uncertainty based AF to ascertain the most certain samples belonging to each source class. The inversion of the lists allows to prioritize the confident predictions for each source class for selection. This makes the diversification procedure more effective in that samples which are predicted as different source class with high certainty are more likely to be different. Here, we note the inverted list as $ent_{\text{invsort}}(\mathbb{D}_T^U)$, $least_{\text{sort}}(\mathbb{D}_T^U)$ and $ent_{\text{sort}}(\mathbb{D}_T^U)$ created by inversion respectively of lists $ent_{\text{sort}}(\mathbb{D}_T^U)$, $least_{\text{invsort}}(\mathbb{D}_T^U)$ and ent_{invsort} as previously defined. Our diversification procedure, named div, as explained in Figure 3.2 and with the pseudo code presented in Algorithm 1 is then performed to select samples.

In the following, we note ent_{inv}^{div}, ms_{inv}^{div} and ls_{inv}^{div}, the resulting set obtained by applying our diversification strategy div respectively on inverted lists $ent_{\text{invsort}}(\mathbb{D}_T^U)$, $least_{\text{sort}}(\mathbb{D}_T^U)$ and $ent_{\text{sort}}(\mathbb{D}_T^U)$.

Algorithm 1 Diversification algorithm

U : a list of unlabeled samples
top : a dictionary containing top prediction source class for all samples in U
b : budget of samples to be selected
procedure div(U, top, b)
    Build L : a list of selected samples from U of length b
    while len(L)$\leq b$ **do**
        seenclasses = empty list : reinitialize memory of source classes
        for each item i in U **do**
            $topsourceclass =$ top[i] :predicted source class for sample $U[i]$
            if $topsourceclass$ not in $seenclasses$ **then**
                if i not in L **then**
                    add sample i in L
                    add $topsourceclass$ in $seenclasses$
                end if
            end if
        end for
    end while
    $L = L[0 : b]$
    return L
end procedure

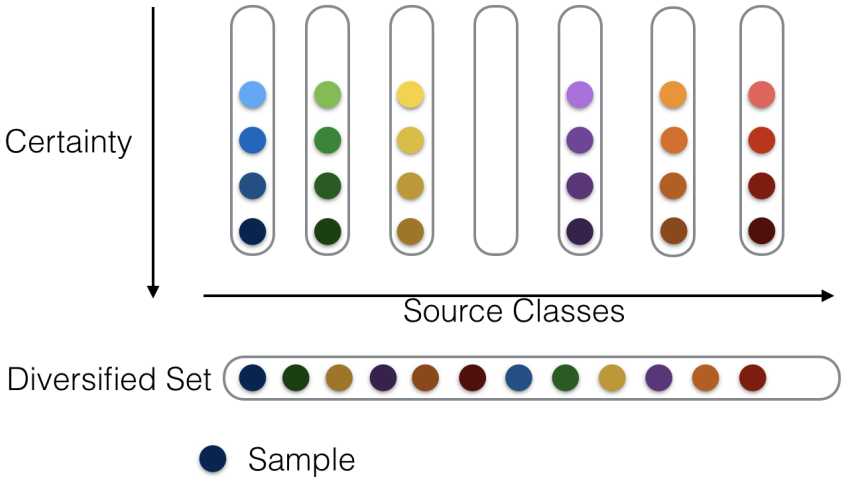


Figure 3.2 – Diversification procedure. Samples are first assigned to their predicted source class and then sorted according to the considered certainty criteria. This creates a list of samples for every source class ordered in terms of certainty of belonging to the class. In the figure, we represent certain samples by using darker shades. Note that it is possible that a source class is not the top predicted class for any target sample and thus does not get any sample assigned to it. The selection of samples is performed iteratively over the source classes, selecting one example per source class, till the budget is filled. If the active learning budget is larger than the number of classes of the source domain, the memory is reset each time all $\mathcal{N}_S$ classes were seen. Thus a diverse set of images is selected by giving equal representation to samples from all the source classes.

3.3.2 . Adding the Balancing component

Here, we introduce a sample balancing step which reduces the propagation of imbalance from the unlabeled to the labeled dataset. A part of the labeling budget is annotated with a classical acquisition approach. A criterion which depends on the budget and on the degree of imbalance in the labeled dataset created so far is proposed to switch toward the balancing step as shown is Figure 3.3. The switch from classical AL selection to balancing step is done to ensure a good balance between discovery and balancing steps of AL. If switching is done too early, balancing is applied to a large number of samples but the number of found classes is likely to be low. Inversely, if the class discovery step is too long, a larger number of classes might be discovered but at the expense of significant imbalance in $\mathbb{D}_T^L$. The switch between the two AL steps needs to be linked to the imbalance profile of the target dataset. The classes are divided into under-represented or over-

represented depending on whether they have less or more samples than the average number of samples m/n_{cm}, where n_{cm} is the number of classes discovered after selection of m samples. It is activated using the following criteria :

$$b - m <= c_{ur} \times (\mu(or) - \mu(ur)) \tag{3.5}$$

with c_{ur} - the number of under-represented classes ; $\mu(or)$ and $\mu(ur)$ - the mean number of samples for under- and over-represented classes when m samples were labeled manually in the current $\mathbb{D}_T^L$

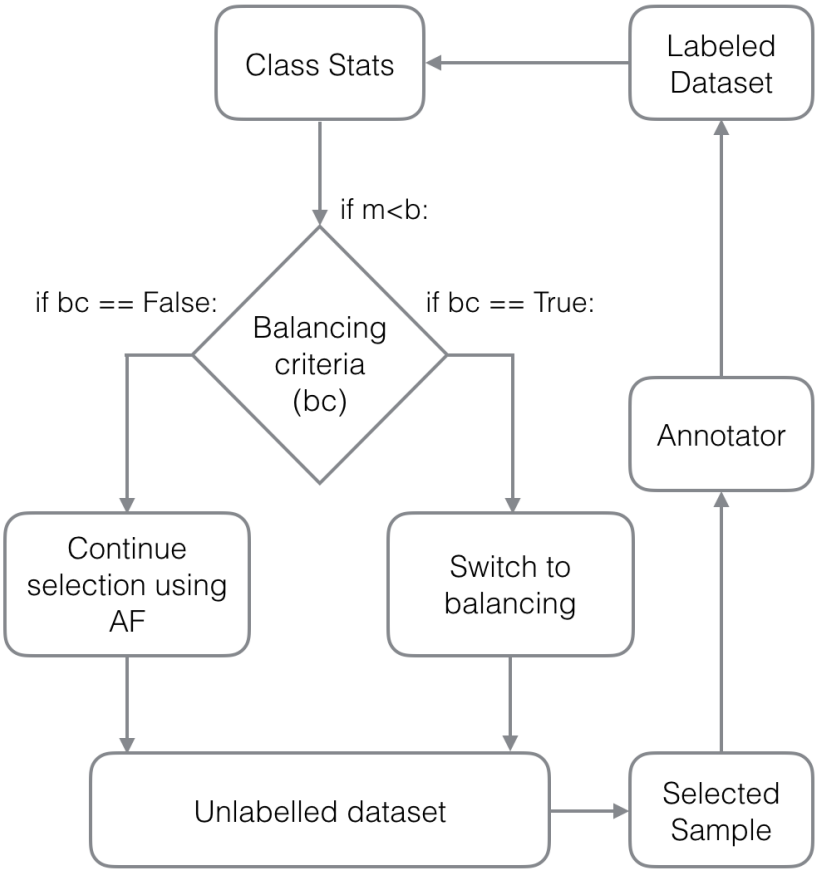


Figure 3.3 – Balancing criteria. The class statistics are computed after each selection. The switch is activated based on the imbalance incorporated in the dataset and the budget left to mitigate this imbalance. Once the balancing criteria is met, the selection of samples proceeds via the balancing algorithm.

For every m value, Equation 3.5 tests if there are enough samples left until b to fill in the gap between the samples of under-represented and over-represented classes. The stronger the imbalance of a dataset, the earlier the switch will be

activated. Note that Equation 3.5 is likely to have little influence for balanced datasets since the switch will be activated very late in the AL process.

Ideally, all samples labeled between m and b would be attributed to under-represented classes in order to have a completely balanced distribution of class samples. In practice, even if under-represented classes are favored during balancing, some imbalance will subsist because :

1. under-represented classes simply do not have enough samples in $\mathbb{D}_T^U$
2. some of the samples attributed during balancing will be directed towards other classes than the intended ones.

Thus once the balancing criteria is met, a given minority class c represented by n_c samples is given a maximum of $b/n_{cm} - n_c$ attempts for balanced selection. This allows all the under-represented classes to have an equal chance of achieve average number of samples per class.

The balancing algorithm is shown in Figure 3.4. We start by prioritizing under-represented classes which have the lowest number of associated samples. Samples from $\mathbb{D}_T^U$ are represented in feature space $\mathbb{R}^d$ provided by the initial model $\mathcal{M}_S$. The mean feature representation is computed for each class using its manually labeled samples in $\mathbb{D}_T^L$. Given the targeted rarest class C_{ur}^{min}, we propose the next sample for labeling using :

$$x_{next} = \min_{\forall i \in \{1, n_T - m\}} \left(\frac{d(\mu(F_S(C_{ur}^{min})), F_S(x_i))}{\max_{\forall j \in \{1, c_{or}\}} (d(\mu(F_S(C_j)), F_S(x_i))} \right) \tag{3.6}$$

with x_i any of the $(n_T - m)$ unlabeled target samples at moment m ; $d(.,.)$ L2-distance in the feature space $\mathbb{R}^d$; c_{or} is the number of over-represented classes ; $\mu(F_S(.))$ - mean features of a class as represented by its samples in the current labeled subset $\mathcal{D}_T^L$.

The numerator in eq. 3.6 favors unlabeled samples which are close to the target class C_{ur}^{min}. The denominator favors samples which are furthest away from any majority class. The imbalance profile of the labeled subset $\mathbb{D}_T^L$ and the mean representations of its known classes are updated after each manual labeling.

3.4 . Experiments

3.4.1 . Training Strategies

The training of a model $\mathcal{M}_T$ over the manually labeled subset $\mathbb{D}_T^L$ can be done by transferring deep features from $\mathcal{M}_S$ or by fine-tuning this model. The first option seems preferable for small AL budgets because fine-tuning a deep architecture might be suboptimal or even impossible. Transfer is implemented using a classical approach [185] which learns shallow classifiers over the features provided by the feature extractor $\mathcal{F}_S$. Inversely, fine-tuning becomes viable if b is larger or if source and target domains are distant from one another. CNN models are

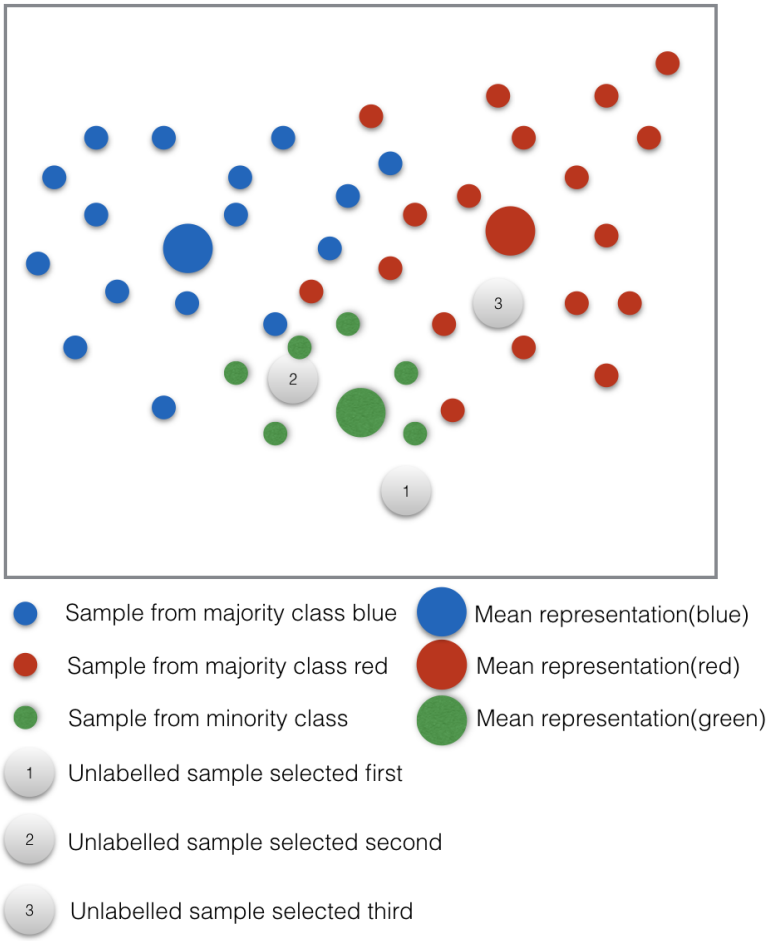


Figure 3.4 – Balancing algorithm applied for minority class (green). The algorithm calculates the mean distance of unlabeled sample to each class in the feature space $\mathbb{R}^d$. The mean position of each class is shown in the figure with bigger circles for simplification. The numbering in the unlabelled samples (larger circles in grey) denotes the order of selection. The selection allows to give maximum chance of samples belonging to the minority class. Thus sample which is closest to the mean representation of minority class and furthest away from the closest majority class is prioritized.

shown to be particularly prone to imbalance and provide prediction scores biased towards majority classes [23]. Following the conclusions of this prior work, a post processing based on prior probabilities is used to calibrate the scores and improve overall accuracy. The choice between the two strategies is done via cross-validation over $\mathbb{D}_T^L$. 10-folds are created and we create both shallow classifiers and fine tuned

Dataset	Class	Images	Mean(μ)	Std(σ)	ir
Food-101	101	22956	227.28	180.31	0.793
CIFAR-100	100	17168	171.68	126.98	0.740
IMN-100	100	18558	185.58	137.16	0.739
MIT-67	67	14281	213.15	168.16	0.789

Table 3.1 – Dataset statistics. ir is the imbalance ratio

models for each fold. Accuracy is averaged over all folds and the strategy which has better performance is selected.

3.4.2 . Datasets

The proposed methods are evaluated on four imbalanced datasets and we consider ILSVRC [193] as source domain. A method [47] to generate pseudo-label for unlabeled images from diverse target domain using a model trained on source domain, shows the viability of using $ILSVRC$ to provide rich high-level representations. We test our methods on four publicly available dataset.

— $Cifar100$ [123] is designed for coarse-grained object classification.
— $Food-101$ [21] is focused on fine-grained food recognition
— $MIT-67$ MIT-Indoor-67 [182] is designed for indoor scene recognition.
— $IMN-100$ is a subset of ImageNet which includes fine-grained classes (i.e. ImageNet leaves). Note that the intersection between $IMN-100$ and $ILSVRC$ is empty.

These datasets are used to evaluate transfer between a large source dataset and target datasets which were created using a protocol different from that of ILSVRC. In addition, we create IMN-100, a subset of randomly selected 100 leaf classes from ImageNet which are not present in ILSVRC. IMN-100 is created to test transfer among classes from the same large collection of images. A common imbalance induction procedure was applied to all datasets using a target imbalance ratio to guide the pruning process. The imbalance ratio is defined as $ir = \frac{\sigma}{\mu}$, with σ standard deviation and μ the mean of images per class in the dataset. The main statistics of the obtained datasets are provided in Table 3.1. Similar imbalance ratio was obtained across datasets to facilitate comparability of results. Imbalance induction process was guided to attain imbalance ratio present in the full ImageNet dataset, which is 0.813 in the 4 target dataset, by transferring the class distribution of ImageNet dataset to the target dataset. Binning is performed on number of images per classes in the ImageNet dataset. The number of bins is set to the number of classes present in the target dataset. Thereafter, the mean of each bin is normalized and multiplied to the mean number of images per class in target dataset to give the images in target imbalanced dataset.

3.4.3 . Implementation Details

The Pytorch [176] pretrained ResNet-18 model is used as $\mathcal{M}_S$. The choice of this model is guided by two criteria : (1) the AL labeled subsets are small and

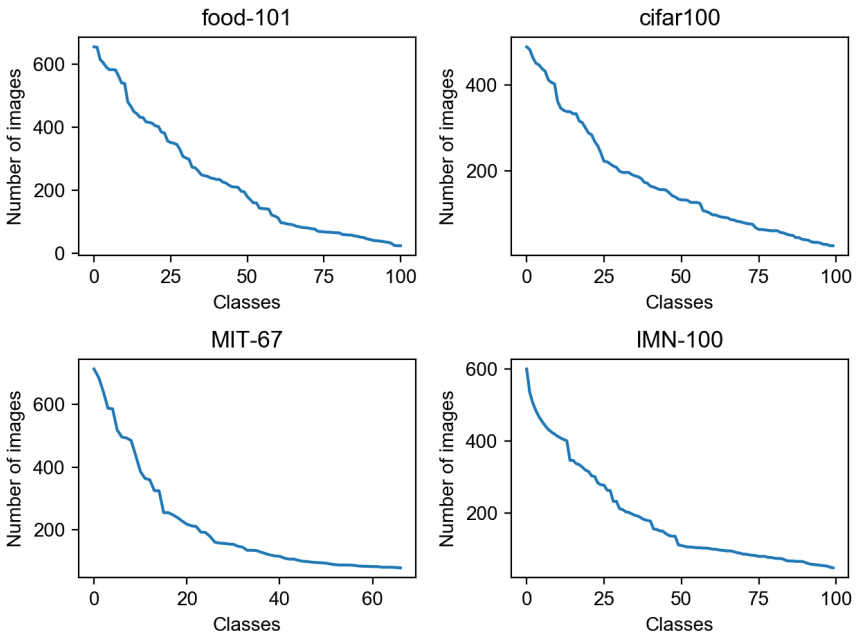


Figure 3.5 – Class imbalance in datasets

deeper models might not converge and (2) the number of experiments to run is large and a relatively quick training is needed. A usual fine-tuning strategy is applied when CNNs are used to create $\mathcal{M}_T$ over the labeled subset $\mathbb{D}_S^L$. Parameters of the source training are kept, except for the initial learning rate which is divided by 10. Linear SVMs from scikit-learn [178] are used to create shallow model when transfer learning is used. Their parameters are optimized using 10-fold cross validation over the labeled subset $\mathbb{D}_S^L$. The choice between SVMs and CNNs to create AL models is done by cross-validation, as explained in Subsection 3.4.1.

Evaluation Methodology

The size of the budget b is the main criterion used to evaluate the performance of active learning methods [199, 198, 15] and we test $b = \{500, 1000, 2000\}$ for each of them. We present results with a range of existing AL acquisition functions and their modified versions described in Subsection 3.2.1. Five runs are launched for non-deterministic acquisition functions ($random$ and $core$) and their accuracy is averaged to prevent accuracy bias. AL performance is evaluated before and after balancing. We also provide details about the number of classes discovered by each $\mathcal{AF}$ and the associated imbalance ratio.

The evaluation measure of individual configurations is top-1 accuracy. It is calculated as an average over the entire set of classes N_t represented in the test

Dataset	Food-101			CIFAR-100			IMN-100			MIT-67			G_{AL}
Budget	500	1000	2000	500	1000	2000	500	1000	2000	500	1000	2000	
$random$	**23.02**	**30.63**	**38.68**	27.31	33.66	39.78	47.24	56.62	63.87	34.99	44.56	**53.33**	-0.792
ent	14.19	20.44	29.26	12.13	17.31	25.18	16.99	24.62	37.58	25.36	31.72	41.20	-1.308
ms	8.49	14.31	28.48	23.70	25.25	35.76	28.46	41.29	38.98	28.91	34.64	46.50	-1.159
lc	15.44	23.45	33.06	15.28	20.79	27.74	21.79	32.09	43.77	27.20	34.68	45.44	-1.191
ent_{inv}	8.84	15.55	26.69	24.19	30.29	34.78	27.83	41.44	38.71	28.87	37.99	42.12	-1.155
ent^{div}	13.93	20.24	30.34	23.96	29.35	35.97	24.25	42.99	55.45	27.07	39.01	44.35	-1.077
ent^{div}_{inv}	19.71	25.60	34.11	**32.13**	**38.94**	**43.94**	53.65	61.21	66.79	39.17	**46.79**	52.09	**-0.739**
ms^{div}_{inv}	16.05	24.26	32.62	24.61	31.46	39.13	39.47	51.68	61.02	31.46	40.99	49.13	-0.928
lc^{div}_{inv}	19.13	24.66	33.62	32.62	38.46	43.52	**55.27**	**61.89**	**66.80**	**39.48**	45.89	51.42	-0.742
$core$	20.07	26.35	34.17	30.04	36.34	42.18	49.84	56.42	63.87	37.10	46.08	52.31	-0.790
$Full$	65.85			59.49			70.20			72.43			-

Table 3.2 – Accuracy of the acquisition functions from Subsection 3.2.1 before balancing. $random$ and $core$ are non deterministic and their performance is averaged over five runs. Best results are presented in bold.

set since the objective is to evaluate the capacity of each AL method to deal with imbalance. Let the total number of classes found by an acquisition function be N_f and the average accuracy over these classes be ACC_f. The final accuracy ACC_t of a configuration is calculated over all the classes which could be discovered using :

$$ACC_t = Acc_f \frac{N_f}{N_t} \tag{3.7}$$

Taking all classes into consideration, even if some of them are not discovered during AL acquisition, is necessary because our objective is to evaluate accuracy over the complete task. The merits of the different methods tested are only comparable if tested for all classes which could be discovered.

Since the number of configurations for each $\mathcal{AF}$ is important, we also present a summarized evaluation of performance. Inspired by recent works such as [187, 212], we propose a global performance score in Equation 3.8.

$$G_{AL} = \frac{1}{c} \times \sum_{i=1}^{c} \frac{acc_i - acc_{full}}{acc_{max} - acc_{full}} \tag{3.8}$$

where : c - number of configurations tested ; acc_i - top-1 score for each configuration (individual values of each row of Table 3.2 and Table 3.3 ; acc_{full} - the upper-bound accuracy of the dataset ($full$ accuracy corresponds to fine-tuning a model for each full imbalanced dataset with ILSVRC as source dataset, followed by score calibration with prior class probabilities as done in [23]) ; acc_{max} - the maximum theoretical value obtainable ($acc_{max} = 100$ here).

G_{AL} measures the performance gap between methods which use a partial labeling of data and an upper-bound which exploits a fully labeled dataset. The denominator is introduced to avoid a disproportionate influence of individual datasets [212]. G_{AL} has a negative value and the closer its value to zero, the better the method is.

Dataset	Food-101			CIFAR-100			IMN-100			MIT-67			G_{AL}
Budget	500	1000	2000	500	1000	2000	500	1000	2000	500	1000	2000	
$random$	**23.53**	**30.52**	37.95	28.86	37.29	44.32	53.79	62.59	68.31	42.36	54.14	60.16	-0.653
ent	19.10	27.06	34.43	24.07	33.82	41.20	41.19	57.65	65.47	34.75	51.68	60.16	-0.792
ms	17.98	29.61	35.40	25.44	35.18	41.71	45.57	51.56	65.73	40.52	48.62	57.17	-0.784
lc	19.59	26.70	37.20	26.68	36.70	40.13	43.03	59.32	67.45	41.30	51.23	59.34	-0.744
ent_{inv}	18.06	28.81	35.62	25.89	34.06	41.87	44.48	57.45	64.08	36.25	49.33	58.15	-0.785
ent^{div}	20.08	26.82	33.57	24.43	34.26	43.20	42.25	55.53	63.33	38.99	51.83	60.01	-0.783
ent_{inv}^{div}	23.20	27.43	**38.00**	**34.32**	**40.78**	45.34	**56.98**	**64.12**	68.21	**47.80**	53.74	60.39	**-0.612**
ms_{inv}^{div}	20.51	27.91	37.50	27.40	37.32	**45.70**	50.48	60.75	66.12	44.67	52.42	59.12	-0.690
lc_{inv}^{div}	21.77	28.71	36.16	32.21	39.92	45.13	55.55	64.05	**68.86**	45.34	51.79	**61.06**	-0.637
$core$	20.84	28.21	37.44	32.68	39.70	44.43	54.57	62.14	67.97	46.42	**54.34**	60.46	-0.640
$Full$	65.85			59.49			70.20			72.43			-

Table 3.3 – Accuracy of the acquisition functions from Subsection 3.2.1 after balancing. $random$ and $core$ are non deterministic and their performance is averaged over five runs. Best results are presented in bold.

	Dataset	Food-101			CIFAR-100			IMN-100			MIT-67			$Average$
Budget		500	1000	2000	500	1000	2000	500	1000	2000	500	1000	2000	
$random$	Classes	87.8	98.2	100.6	91	97	99	92.8	99	100	66	67	67	88.8
	ir	**0.936**	**0.849**	**0.820**	0.837	0.785	0.757	0.864	0.798	0.772	**0.857**	**0.796**	**0.784**	**0.821**
ent	Classes	77	90	99	64	77	91	58	80	90	54	65	66	75.917
	ir	1.758	1.528	1.328	2.480	2.079	1.556	2.947	2.304	1.735	1.280	1.148	1.031	1.765
ent_{inv}^{div}	Classes	85	92	100	97	98	99	99	99	100	64	67	67	89
	ir	1.292	1.267	1.111	**0.723**	0.710	0.706	0.587	**0.550**	**0.515**	0.928	0.914	0.823	0.844
lc_{inv}^{div}	Classes	84	92	99	95	98	99	99	100	100	63	65	67	88.41
	ir	1.235	1.226	1.067	0.732	**0.686**	**0.683**	**0.571**	0.573	0.524	0.898	0.887	0.837	0.827
$core$	Classes	84.8	95	100	93	99	100	98	100	100	65.2	67	67	89.03
	ir	1.266	1.228	1.170	0.926	0.831	0.767	0.844	0.774	0.754	0.918	0.853	0.820	0.929
$Full$	Classes	101			100			100			67			92
	ir	0.793			0.740			0.739			0.789			0.765

Table 3.4 – Number of classes found and imbalance ratio for the main acquisition methods before balancing. The number of classes is not an integer for $random$ and $core$ because these methods are not deterministic and their performance is averaged over five runs.

	Dataset	Food-101			CIFAR-100			IMN-100			MIT-67			$Average$
Budget		500	1000	2000	500	1000	2000	500	1000	2000	500	1000	2000	
$random$	Classes	90.6	97.4	100.4	90.8	96.6	99.4	93.4	98	99.6	65.8	67	67	88.83
	ir	**0.803**	**0.820**	**0.841**	0.750	0.677	0.635	0.491	0.357	0.241	0.586	**0.297**	**0.264**	0.563
ent	Classes	85	98	100	88	95	100	86	96	100	63	64	67	86.83
	ir	1.236	1.035	1.058	0.998	0.891	0.815	1.511	0.975	0.821	0.789	0.476	0.363	0.914
ent_{inv}^{div}	Classes	88	98	101	95	100	100	95	100	100	64	67	67	89.58
	ir	0.986	0.976	0.850	**0.587**	**0.559**	0.655	**0.368**	0.377	**0.187**	**0.449**	0.434	0.341	**0.564**
lc_{inv}^{div}	Classes	85	98	100	95	98	100	93	100	100	62	66	67	88.66
	ir	0.849	0.865	0.908	0.710	0.614	**0.613**	0.420	**0.337**	0.210	0.522	0.405	0.352	0.567
$core$	Classes	89.8	96.8	100.8	91.4	99	99.800	97	99.4	100	65	66.6	67	89.38
	ir	0.943	0.956	0.894	0.713	0.689	0.662	0.568	0.417	0.289	0.450	0.373	0.323	0.606
$Full$	Classes	101			100			100			67			92
	ir	0.793			0.740			0.739			0.789			0.765

Table 3.5 – Number of classes found and imbalance ratio for the main acquisition methods after balancing. The number of classes is not an integer for $random$ and $core$ because these methods are not deterministic and their performance is averaged over five runs.

3.4.4 . Performance of Acquisition Functions

A first finding provided by Table 3.2 is that existing $\mathcal{AF}$ are not well adapted for the single stage AL. Their performance, as measured by G_{AL} and for individual configurations, is lower than that of random sampling. This is notably the case for uncertainty-based functions whose G_{AL} is consequently lower compared to random sampling. Even the recent $core$ method has global performance equivalent to that of random sampling. This is somewhat expected since the direct application of $\mathcal{AF}$ on a different source domain is not suited for our task.

A second important finding is that the proposed $\mathcal{AF}$ adaptations are efficient since performance is improved for all uncertainty-based methods when diversification is applied to their inversed definitions as discussed in Subsection 3.2.1. The performance gain is particularly interesting for the modified versions of entropy ent^{div}_{inv} and least confidence (lc^{div}_{inv}) which gain 0.57 and 0.45 G_{AL} point respectively and are globally better than that of $random$. As shown by the intermediate result obtained for ent_{inv} and ent^{div}, both the shift from uncertain to representative images and the use of the diversification scheme-based on the predictions of the pretrained model are beneficial. Note that we tried to apply the diversification procedure to the $core$ too but results were inconclusive. This negative finding is probably explained by the fact that geometric-based functions are in the feature space, and diversification is applied to the classifier predictions.

The analysis of individual configurations, ent^{div}_{inv} and lc^{div}_{inv}, indicates that they are clearly better compared to $random$ for CIFAR-100 and IMN-100 and also for the lower budgets of MIT-67. Gains are more important for lower budgets, i.e. the most difficult and interesting AL configurations since they allow a larger reduction of the labeling effort. At higher budgets, for instance at $b = 2000$, the accuracy of the different methods as much closer. Interestingly, $random$ is clearly the best method for Food-101. This behavior underlines a limitation of deep representation transferability, regardless of its implementation via transfer learning with shallow classifiers or by fine-tuning the initial model. The result is explained by the larger visual gap between Food-101 and ILSVRC which translates into a significantly higher difference between AL scores and the performance on the full dataset. We provide further analysis of transferability in Subsection 3.4.6.

In Table 3.4, we complement the analysis of accuracy with a presentation of the number of classes discovered by each method and the standard deviation in the distribution of labeled samples. An ideal method would discover all classes and have a standard deviation as close as possible to zero in order to give all classes similar chances of being recognized. Only the main methods from Table 3.2 are kept here. Results are rather well correlated to accuracy, with ent^{div}_{inv} having the best behavior for CIFAR-100 and IMN-100 and $random$ being best for Food-101. The low accuracy of classical entropy is explained by its poor behavior both in terms of class discover and of imbalance ratio. Interestingly, while $random$ samples are more balanced for MIT-67 compared to ent^{div}_{inv}, accuracy remains better for the

latter method. This is probably explained by the fact that the labeled samples are more representative of each class for ent^{div}_{inv} compared to a random selection. The results in Table 3.5 also validate our hypothesis that the application of acquisition functions worsens the global imbalance of $\mathbb{D}^U_T$. None of the acquisition functions has imbalance lower than that of the full imbalanced datasets. This justifies the need for a balancing step during the acquisition process.

3.4.5 . Influence of Balancing

Balancing provides a consequent improvement for all $\mathcal{AF}$. The G_{AL} scores after balancing (Table 3.5) are clearly better than those obtained before balancing (Table 3.4). The G_{AL} score for $random$ moves from -0.792 to -0.653, while that of ent^{div}_{inv} goes from -0.739 to -0.612. lc^{div}_{inv} remains second best but with an increased gap compared to ent^{div}_{inv}. We note also that balancing improves performance of acquisition function for the Food-101 dataset. In particular, ent^{div}_{inv} is on par with $random$ for $b = 500$ and $b = 2000$ but still lags behind for $b = 1000$. This result indicates that even balancing is useful to some extent even when feature transferability is low.

The comparison of imbalance ratios before and after balancing provided in Tables 3.4 and Table 3.5 shows that the proposed procedure is useful. The reduction of imbalance contributes to the improvement of accuracy compared to the case when no balancing is applied. The average imbalance ratio for $random$ and ent^{div}_{inv} is 0.821 and 0.844 without balancing compared to 0.563 and 0.564 with balancing to be compared with 0.765 for the full imbalanced datasets.

The balancing process also provides a slight increase of the number of classes discovered, which is another important factor which contributes to accuracy. This can be explained by the fact that when switching between acquisition modes, the acquisition strategy changes and a different subspace of the feature space is explored.

3.4.6 . Analysis of Transferability

The distance from source to target domains conditions the success of transfer learning [185]. The larger this distance is, the higher the chances for transfer to be inefficient are. The differences of accuracy between the training with the full dataset and with AL methods provided in Tables 3.2 and 3.3 indicate that the distance between the ILSVRC source is highest for Food-101. We deepen this simple estimation of transferability in Figure 3.6. It shows the mapping of top-1 predictions for the training images in the target datasets over the classes of the source dataset. Transfer is likely to be successful if the mapping encompasses a large number of ILSVRC classes and is rather balanced. Such a distribution would indicate that the target domain is richly represented in the source domain. Inversely, a distribution concentrated on a small number of classes indicates that the target is poorly represented and transfer would be less likely to succeed. The distributions from Figure 3.6 are directly comparable for Food-101, CIFAR-100,

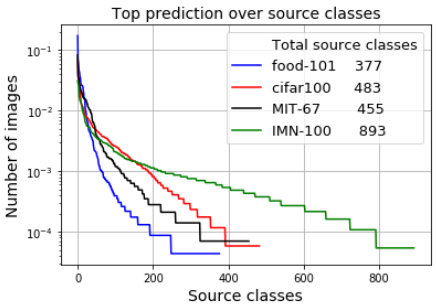


Figure 3.6 – Distribution of number of target dataset images predicted per source class. Source classes are ranked from left to right from most to least frequent. To facilitate comparability, the raw number of predictions is divided by the size of each target dataset. Best viewed in color

Dataset	Food-101			CIFAR-100			IMN-100			MIT-67			G_{AL}
Budget	500	1000	2000	500	1000	2000	500	1000	2000	500	1000	2000	
$random$	**26.73**	**34.75**	**43.20**	34.78	**43.81**	**51.33**	**56.75**	**65.81**	**71.50**	**48.39**	**57.95**	**64.47**	**-0.538**
ent_{inv}^{div}	23.45	28.15	36.72	**35.75**	42.48	49.17	53.39	62.47	69.10	48.09	55.53	62.18	-0.627
ls_{inv}^{div}	23.33	28.04	36.96	35.99	43.42	49.76	55.46	62.07	69.67	46.67	54.63	62.78	-0.620
$core$	22.43	28.55	37.82	32.34	41.65	49.13	51.32	59.34	67.29	45.99	55.05	62.78	-0.662
$Full$	68.53			63.02			72.89			65.47			-

Table 3.6 – Accuracy of the acquisition functions with balanced dataset before balancing. $random$ and $core$ are non deterministic and their performance is averaged over five runs. Best results are presented in bold.

IMN-100 are directly comparable because these datasets have a nearly identical number of classes. The distribution is the least balanced for Food-101, followed by CIFAR-100 and IMN-100. This mirrors the accuracy reported for each dataset in Tables 3.2 and 3.3. MIT-67 has fewer classes and its distribution is naturally tighter. However, it is still more evenly distributed than that of Food-101. This analysis underlines that ILVSRC is a good source domain dataset.

3.4.7 . Analysis with Balanced datasets

The balancing step introduced in Section 3.3.2 is intended for imbalanced datasets. However, it is interesting to also test its behavior, as well as that of the proposed acquisition functions, for balanced datasets. Tests are performed over balanced subsets of the datasets which include a number samples per class comparable to that of imbalanced versions. There are 200 images per class for Food-101, CIFAR-100 and IMN-100 is set to 200 and 80 for MIT-67. The number of images is lower for the latter dataset because its least represented classes include 80 images. The performance of diversification based AF are comparable, with $random$ being

Dataset	Food-101			CIFAR-100			IMN-100			MIT-67			G_{AL}
Budget	500	1000	2000	500	1000	2000	500	1000	2000	500	1000	2000	
$random$	**27.49**	**36.18**	**44.11**	35.76	45.30	**51.94**	58.57	**66.67**	**71.42**	**51.79**	**59.46**	**64.80**	**-0.502**
ent_{inv}^{div}	26.10	29.33	40.52	**38.48**	45.18	50.62	59.70	65.32	70.31	48.95	58.67	64.65	-0.544
ls_{inv}^{div}	25.18	31.53	40.91	37.89	**46.09**	51.05	**59.96**	65.69	70.40	50.75	57.10	64.28	-0.536
$core$	24.83	32.23	41.42	35.03	43.50	50.42	55.71	64.44	69.84	49.72	58.30	63.39	-0.568
$Full$	68.53			63.02			72.89			65.47			-

Table 3.7 – Accuracy of the acquisition functions with balanced dataset after balancing. $random$ and $core$ are non deterministic and their performance is averaged over five runs. Best results are presented in bold.

	Dataset	Food-101			CIFAR-100			IMN-100			MIT-67			$Average$
Budget		500	1000	2000	500	1000	2000	500	1000	2000	500	1000	2000	
$random$	Classes	100.4	101	101	99.4	100	100	99	100	100	66.6	67	67	91.78
	ir	**0.463**	**0.316**	**0.204**	**0.442**	**0.318**	**0.216**	**0.446**	**0.293**	**0.220**	**0.361**	**0.227**	**0.146**	**0.304**
ent_{inv}^{div}	Classes	87	100	101	97	100	100	99	100	100	66	67	67	90.33
	ir	0.989	0.968	0.776	0.542	0.516	0.440	0.534	0.556	0.462	0.556	0.479	0.340	0.596
lc_{inv}^{div}	Classes	90	99	101	99	100	100	100	100	100	66	67	67	90.75
	ir	0.990	0.969	0.778	0.525	0.496	0.416	0.510	0.557	0.457	0.599	0.494	0.350	0.595
$core$	Classes	92.8	98.6	100.8	98.8	100	100	98.8	99.8	100	67	67	67	90.88
	ir	0.957	0.860	0.763	0.578	0.542	0.465	0.707	0.654	0.578	0.627	0.488	0.359	0.631
$Full$	Classes	101			100			100			67			92
	ir	0			0			0			0			0

Table 3.8 – Number of classes found and imbalance ratio for the main acquisition methods with balanced datasets before balancing. The number of classes is not an integer for $random$ and $core$ because these methods are not deterministic and their performance is averaged over five runs.

	Dataset	Food-101			CIFAR-100			IMN-100			MIT-67			$Average$
Budget		500	1000	2000	500	1000	2000	500	1000	2000	500	1000	2000	
$random$	Classes	99.4	101	101	98.4	100	100	97.6	100	100	66.8	67	67	91.52
	ir	**0.415**	**0.225**	**0.104**	0.321	**0.186**	**0.095**	0.280	**0.144**	**0.084**	**0.174**	**0.092**	**0.056**	**0.181**
ent_{inv}^{div}	Classes	99	100	101	98	100	100	98	100	100	67	67	67	91.42
	ir	0.661	0.802	0.428	0.261	0.192	0.134	0.261	0.192	0.134	0.302	0.171	0.122	0.305
lc_{inv}^{div}	Classes	98	100	101	99	100	100	99	100	100	67	67	67	91.50
	ir	0.721	0.542	0.446	**0.192**	0.191	0.149	**0.192**	0.191	0.149	0.244	0.167	0.143	0.277
$core$	Classes	97	101	101	98.3	100	100	97.8	99.6	100	66.6	67	67	91.27
	ir	0.642	0.464	0.291	0.315	0.213	0.168	0.363	0.241	0.204	0.331	0.181	0.141	0.296
$Full$	Classes	101			100			100			67			92
	ir	0			0			0			0			0

Table 3.9 – Number of classes found and imbalance ratio for the main acquisition methods with balanced datasets after balancing. The number of classes is not an integer for $random$ and $core$ because these methods are not deterministic and their performance is averaged over five runs.

Dataset	Food-101			CIFAR-100			IMN-100			MIT-67			G_{AL}
Budget	500	1000	2000	500	1000	2000	500	1000	2000	500	1000	2000	
$random$	**23.02**	**30.63**	**38.68**	27.31	33.66	39.78	47.24	56.62	63.87	34.99	44.56	53.33	-0.792
ent_{inv}^{div}	19.71	25.60	34.11	32.13	38.94	43.94	53.65	61.21	66.79	39.17	46.79	52.09	-0.739
$ent_{inv}^{div}+ens$	*19.63*	*26.20*	*33.84*	*34.67*	*42.67*	***47.78***	*58.24*	*63.67*	***69.11***	***43.17***	*46.34*	*55.01*	*-0.672*
ls_{inv}^{div}	19.13	24.66	33.62	32.62	38.46	43.52	55.27	61.89	66.80	39.48	45.89	51.42	-0.742
$ls_{inv}^{div}+ens$	*19.82*	*26.17*	*33.55*	***35.77***	***42.95***	47.23	***60.32***	***64.87***	*68.85*	*42.99*	*47.56*	*53.81*	***-0.663***
$core$	20.07	26.35	34.17	30.04	36.34	42.18	49.84	56.42	63.87	37.10	46.08	52.31	-0.790
$core+ens$	*19.90*	*26.34*	*33.86*	*31.95*	*38.29*	*46.10*	*54.08*	*59.90*	*66.41*	*38.73*	***48.79***	***55.62***	*-0.723*
$Full$	65.85			59.49			70.20			72.43			-

Table 3.10 – Accuracy of the acquisition functions with ensemble before balancing. We take the results for main methods from Table 3.2. For ensemble, we add $+ens$ to method names and present the results in italics to improve readability. Note that $random$ is not influenced by ensembles and is the same as in Table 3.2.

most effective especially at higher budgets and $core$ the least effective method, as reported in Table 3.6.

Somewhat surprisingly, balancing is beneficial for all acquisition functions as shown in Table 3.7. Even though the tested datasets are globally balanced, the selection of a subset for annotation results in an imbalanced distribution. Imbalance is naturally larger for lower budgets because subset is least representative of the entire distribution. Accuracy gains are generally between two and three points for lower budgets, which are most interesting in AL since they require the lowest annotation effort. Also interesting, the global performance of ent_{inv}^{div} and ls_{inv}^{div} becomes closer to that of $random$. For the lowest budget, ent_{inv}^{div} and ls_{inv}^{div} are more competitive than $random$ after balancing for CIFAR-100 and IMN-100, the two datasets with best transferability from the source. The comparison of imbalance ratio and classes found in Table 3.8 and 3.9 shows that none of the AF methods finds a perfectly balanced subset for manual annotation. However, the degree of imbalance is considerably reduced after the application of balancing. For instance, it is more than halved for ent_{inv}^{div} and ls_{inv}^{div} when applied to CIFAR-100 and IMN-100 for all three AL budgets. The number of discovered classes is higher than that reported for imbalanced datasets. This is intuitive since class discovery is simpler when classes are balanced and the odds to find representatives of each class are comparable.

3.4.8 . Active Learning with Ensembles

The authors of [15] showed that the use of ensembles is beneficial in active learning. They use different snapshots selected during the training process of a CNN to obtain an ensemble of models. Features for the ensemble are obtained by applying a average pooling operator. We use the same methodology here. A ResNet-18 model was trained for 90 epochs, six snapshots were retained every 15 epochs and their features and probabilities were then averaged. The results obtained with the ensemble before and after balancing are reported in Tables 3.10

Dataset	Food-101			CIFAR-100			IMN-100			MIT-67			G_{AL}
Budget	500	1000	2000	500	1000	2000	500	1000	2000	500	1000	2000	
$random$	23.53	**30.52**	37.95	28.86	37.29	44.32	53.79	62.59	68.31	42.36	54.14	60.16	-0.653
ent^{div}_{inv}	23.20	27.43	**38.00**	34.32	40.78	45.34	56.98	64.12	68.21	47.80	53.74	60.39	-0.612
$ent^{div}_{inv}+ens$	***25.26***	*28.77*	*37.00*	***36.67***	*42.77*	***48.54***	***60.76***	*66.49*	***69.98***	*49.07*	*55.75*	*63.98*	*-0.548*
ls^{div}_{inv}	21.77	28.71	36.16	32.21	39.92	45.13	55.55	64.05	68.86	45.34	51.79	61.06	-0.637
$ls^{div}_{inv}+ens$	*22.12*	*28.59*	*34.72*	*35.53*	*42.93*	*48.77*	*59.47*	***67.08***	*69.61*	*45.53*	*54.26*	*63.98*	*-0.57 6*
$core$	20.84	28.21	37.44	32.68	39.70	44.43	54.57	62.14	67.97	46.42	54.34	60.46	-0.640
$core+ens$	*19.77*	*27.18*	*37.56*	*35.31*	***43.26***	*48.38*	*57.47*	*65.39*	*69.41*	***52.10***	***58.68***	***64.56***	***-0.554***
$Full$	65.85			59.49			70.20			72.43			-

Table 3.11 – Accuracy of the acquisition functions with ensemble after balancing. We take the results for main methods from Table 3.3. For ensemble, we add $+ens$ to method names and present the results in italics to improve readability. Note that $random$ is not influenced by ensembles and is the same as in Table 3.3.

and 3.11 respectively. They indicate that the use of ensemble features is indeed effective in most configurations and provides a performance improvement over the non-ensemble counterpart. Further, the findings reported with vanilla features are replicated with ensemble features, with diversification based methods outperforming $random$ in most setting and balancing also providing improvement for all the AF. The best strategy for Food-101, the dataset with lowest transferability from the source model, remains $random$ as this was the case for the experiments with imbalanced datasets.

3.5 . Conclusion

We introduced the single stage AL setting to tackle the cold start problem encountered at the start of AL process. The target dataset is completely unlabelled and number of classes are ascertained after the annotation process. A large external annotated dataset is used to learn a source model. The modified acquisition functions take advantage of a pretrained deep model to find diverse and class-balanced set of samples for manual labeling. We tested the methods in their ability to discover maximum number of classes and to provide the best performance on target domain. The focus was on imbalanced datasets and to limit the propagation to imbalance from unlabeled to labelled set. The probability and feature estimates from the source model is used to implement the diversification and balancing constraints. A diversification method was added to the classical acquisition functions to select samples which give different source class predictions. Further a balancing step was introduced which is activated depending on the imbalance accumulated in the labeled set and the budget left. The balancing step focuses the labeling process on classes which are underrepresented in the annotated subset. Both adaptations have a positive effect as long as features are efficiently transferable between the source model and the target imbalanced datasets. We also show that the proposed me-

thod helps to reduce the imbalance in the selected set. Further, we also tested our methods on balanced datasets and show that the balancing step is beneficial for all acquisition functions. It is particularly the case for datasets with lower global performance.

Obtained results are encouraging and further work can be pursued along three lines. First, new diversification methods for the acquisition functions can be made tested based on using different source models. Second, a pretrained model learned on a larger dataset to ensure transferability toward a larger spectrum of target datasets can be considered. Finally, methods to determine whether representations are transferable between source and target datasets can be investigated. If this is not the case, it becomes preferable to run random sampling followed by balancing instead of more sophisticated acquisition functions.

4 - Iterative active learning for imbalanced datasets

In this chapter, we introduce a new active learning method for imbalanced datasets in the iterative AL setting in which we assume the presence of an initial annotated dataset to kick-start the AL process. This iterative setting is widely studied in the literature of active learning [198, 15]. As discussed in Chapter 3, the effectiveness of single stage AL depends on the transferability between the source and target domain. Further, with a sufficient initial budget the model trained on the target domain becomes effective for the AL task. Thus, the iterative setting would become preferable when we dispose of a sufficient annotation budget.

In the previous chapter, we defined the balancing and diversification constraints for imbalanced datasets in the single stage setting. These objectives also remain important in the iterative setting to mitigate the transfer of imbalance from unlabelled to labeled set. We propose a method which favors samples likely to be in minority classes so as to reduce the imbalance of the labeled subset and create a better representation for these classes. Further, we test three strategies for selection of samples assigned to minority class. Evaluation is done with three imbalanced datasets designed for different visual tasks. Results indicate that the proposed method outperforms competitive baselines as introduced in Chapter 3. In addition to global results, we present an analysis of the method components so as to understand their individual roles.

The outline of the chapter is as follows. The first section 4.1 contains the context and the motivations of our proposed method. In section 4.2, we detail it. We provide an experimental analysis and a discussion on the method in section 4.3. Finally, we provide some conclusions in section 4.4.

4.1 . Motivations

We introduce a new method which tackles imbalance in AL by focusing on samples which are classified as belonging to minority class by the model learned in the previous AL iteration. While simple, this approach has two advantages. First, if the samples are correctly classified as minority classes, the selection of these samples mitigates imbalance in the labeled subset and results in a better representation of the minority classes. Second, if the samples are mis-classified as minority, we hypothesize that these samples have high informative value. Indeed, in this case, the model learns from samples that it previously mis-classified, thereby adding important missing information. It could also help to prune the decision boundaries around the minority classes as the samples mis-classified as minority class are likely to be somewhat in the vicinity of the minority class. We use three selection

strategies to sample from minority class based on uncertainty, certainty or diversity. These intra-class selection strategies mirror the usual AL selection strategies which are applied at the dataset level. The minority status of a class is dynamically assigned after each iteration by updating statistics about the class distribution. The number of samples selected for a minority class depends on the imbalance profile. Note that minority class predictions might not be numerous enough to cover the entire AL iteration. If so, the three intra-class strategies described above become equivalent since all minority samples will be selected. Then, the remaining budget of the iteration will be selected using a classical acquisition function, such as random or margin sampling.

In the last chapter, we showed that the training shallow classifier over the feature representation extracted from pretrained model is preferable to fine-tune all the parameters of the model in the early stage of single-stage AL, particularly if pre-trained features are transferable toward the current task. Beyond the proposal of a new AL method, we provide a comparison of these two learning strategies in iterative AL and propose a combination of them to maximize accuracy.

In the previous chapter, we tackled the cold start in AL for imbalanced datasets using a single-stage scenario. The selection of an initial diversified and balanced sample set was done using a pre-trained model. The main differences between the approach proposed here and the one in previous chapter are : (1) the proposal of a different acquisition method, (2) the use of a more generic iterative AL setting and (3) the adaptation of shallow classifiers to an imbalanced context.

4.2 . Proposed method

Minority classes have weaker representations in the models trained from imbalanced datasets. They need to be prioritized either during training or during post processing to reduce the effect of imbalance [23, 86]. We translate this observation to an iterative AL scenario to propose a simple and efficient method which improves sampling from imbalanced datasets. Minority classes are identified by computing statistics of the class distribution of labeled data points up to the last iteration. This distribution also provides the estimated number of samples needed in the current iteration to remove imbalance for each minority class. The set of candidates for a minority class is made of samples predicted as belonging to it in the last trained model $\mathcal{M}_{k-1}$. Selection of candidate samples can be done to boost certainty, uncertainty or diversity and leads to the different versions of the proposed AF discussed below.

In an iterative AL setting, a total budget b is allocated for manual labeling in t iterations with $\frac{b}{t}$ samples selected in each iteration. The process starts by randomly selecting an initial subset of $\mathbb{D}^U$ for annotation to create the initial labeled dataset $\mathbb{D}_0^L$ with $x_j, y_j \in \mathcal{X} \times \mathcal{Y}$ for $j = [1..\frac{b}{t}]$. Afterwards, at iteration step k, for $k = [1..t-1]$, a batch of samples of size $\frac{b}{t}$ is selected for labeling from

$\mathbb{D}_k^U = \mathbb{D}^U \setminus \mathbb{D}_{k-1}^L$, and added to $\mathbb{D}_{k-1}^L$ to update the labeled subset $\mathbb{D}_k^L$. $\mathbb{D}_k^L$ is then used to learn the model $\mathcal{M}_k$ with parameters θ_k.

At the start of the k^{th} iteration, the number of labeled samples is $s_k = k\frac{b}{t}$ and the objective is to add $\frac{b}{t}$ new samples with priority given to minority classes. We note s_k^c as the number of labeled samples for class c and compute the average number of samples per class $\mu_k = \frac{s_k}{C}$, where C is the number of classes. A class is then considered as minority if $s_k^c < \mu_k$. The maximum number of samples which is allowed for class c during the k^{th} iteration is defined as :

$$m_k^c = \begin{cases} \mu_k - s_k^c, & \text{if } s_k^c < \mu_k \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

Equation 4.1 favors minority classes since they are the only ones which have candidate samples allocated. The set of unlabeled samples associated to class c is given by :

$$\mathbb{D}_c^{U(k)} = \{\forall x \in \mathbb{D}_k^U, if\ P(c^1 = c|x)\} \tag{4.2}$$

, where c^1 is the predicted label for the sample x and $\mathbb{D}_k^U$ is the set of unlabelled samples at iteration k.

If $|\mathbb{D}_c^{U(k)}| > m_k^c$, a selection is needed among the set of samples given by Equation 4.2.

We propose three ways to select samples which are inspired from classical AL objectives as shown in Figure 4.1. We present the methods with margin sampling as the base informative measure, but it can be replaced by any other informative measure.

4.2.1 . Certainty-oriented Minority Class Sampling

It favors the most certain data points from $\mathbb{D}_c^{U(k)}$ using :

$$CMCS = arginvsort_{\forall x \in \mathbb{D}_c^{U(k)}} marg_k(x, \theta_k) \tag{4.3}$$

where $marg_k(x, \theta_k)$ is the margin measure from Equation 2.4 and $arginvsort$ sorts the samples in decreasing order. Note that Equation 4.3 performs a margin sampling at class level instead of dataset level. $CMCS$ thus allows a selection of certain samples for each minority class according to Equation 4.1 for sample allocation to classes.

4.2.2 . Uncertainty-oriented Minority Class Sampling

It favors the most uncertain samples from $\mathbb{D}_c^{U(k)}$ using :

$$UMCS = argsort_{\forall x \in \mathbb{D}_c^{U(k)}} marg_k(x, \theta_k) \tag{4.4}$$

where $marg_k(x, \theta_k)$ the margin sampling criteria from Equation 2.4 and $argsort$ sorts the samples in increasing order. Equation 4.4 favors data points which are predicted under c but are close to other classes. Its objective is inverse compared to that of $CMCS$.

4.2.3 . Diversity-oriented Minority Class Sampling

It aims to select a diversified sample subset for c. Such a subset can be obtained, for instance, by applying the Coreset method [198] from Equation 2.6 to $\mathbb{D}_c^{U(k)}$:

$$DMCS = core(\mathbb{D}_c^{U(k)}, \mathbb{D}_c^{L(k)}) \tag{4.5}$$

where $core$ as defined in Equation 2.5 is applied iteratively to select m_k^c samples from $\mathbb{D}_c^{U(k)}$ using the samples already selected $\mathbb{D}_c^{L(k)}$ for class c, while updating the datasets with every selection.

Auxiliary acquisition functions The proposed sampling process is focused on minority classes. It is possible that, if the imbalance is limited, the number of samples allocated to minority classes is less than the budget. Further, there is no guarantee that there are enough samples predicted under minority classes to treat the imbalance. Minority predicted samples are most likely to be insufficient either at the very beginning or towards the end of the AL process. In the beginning, minority class representations are weak and their samples are likely to be mis-classified in majority classes. Towards the end, there will be simply too few samples left for labeling in minority classes. In such cases if the budget of the k^{th} iteration is not filled entirely, remaining samples can be selected according to any AF. Tests are run using random and margin sampling for these remaining samples. The final forms proposed acquisition functions are noted $DMCS - rand$ and $DMCS - marg$.

4.3 . Experiments

We first describe the experimental setup and the datasets used in evaluation. Then, we discuss the obtained results globally and also present an analysis of the main components of the proposed approach.

4.3.1 . Certainty-diversified sampling

In addition to classical baselines such as margin sampling, coreset and random sampling we also compare the method with the contribution from single stage AL scenario introduced earlier. We introduced this function, abbreviated $cds - bal$ below, in last chapter to deal with the cold start problem in single-stage imbalanced AL. It exploits a pretrained model $\mathcal{M}_S$ to provide the features for its diversification

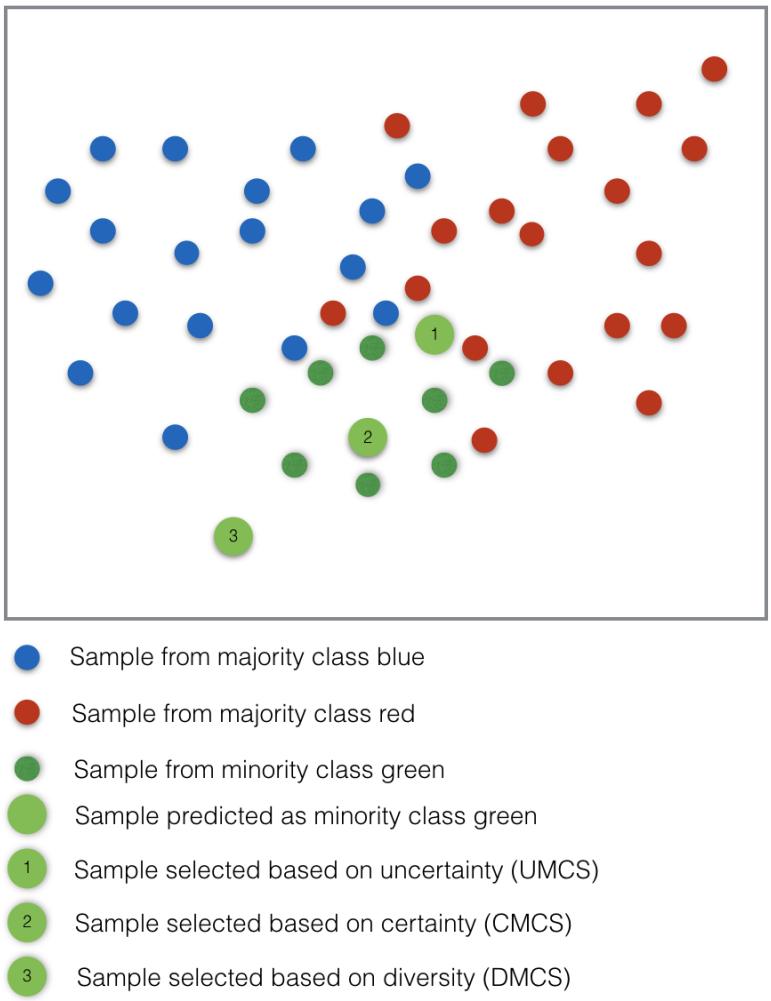


Figure 4.1 – The method prioritises selection for minority classes based on imbalance in the labeled set. For minority class (green) selection is done from the samples predicted (larger circle) as minority class. Three different strategies $CMCS$, $UMCS$ and $DMCS$ are proposed based on selecting the most certain, uncertain or diverse samples respectively. $CMCS$ favors the certain samples and thus is most likely to reduce imbalance. $UMCS$ based on uncertainty selects samples from the decision boundary for minority class and would potentially select more informative samples. $DMCS$ selects diverse samples using the $core$ baselines to create a diverse representation for majority class. Note that if the number of samples classified as minority class is less than or equal to samples allocated to the given minority class, the three variants would select the same samples.

and balancing objectives. Note that $cds - bal$ is deployed in the feature space so as to select samples which minimize their distance to the centroid of a minority class and to maximize the distance to the centroid of the closest majority class. $cds - bal$ is adapted here for usage in an iterative AL setting by selecting the initial subset using random sampling as in other AL acquisition function.

4.3.2 . Setup

We test the proposed approach using an usual iterative AL setting [15, 198, 57]. We set the AL budget to $b = 8000$ and the number of iterations to $t = 16$, including the initial one. The number of samples selected in each iteration is 500.

We use a ResNet-18 architecture [88] for all experiments. The ResNet-18 model, trained over the ILSVRC dataset [193], is used $\mathcal{M}_S$.

AL performance is tested with two training schemes. The first scheme is based on fine tuning as proposed in previous deep AL works [15, 198, 57]. We employ thresholding based on prior class probabilities to reduce the effects of imbalance [23]. This scheme is noted $FT - th$ and is used by default in experiments. $FT - th$ models are trained for 60 epochs with an initial learning rate of 0.01 and a batch size of 32. The Stochastic gradient descent was used with the cross-entropy loss. A learning rate decay of 0.1 was done if the loss plateaus for 10 epochs. The second scheme is inspired from transfer learning and exploits a model pretrained on ILSVRC. It is less frequent in deep active learning but proved useful to tackle cold start problem in Chapter 3. SVMs are trained after each iteration using the features provided by $\mathcal{M}_S$, the pretrained model. Following [50, 258], cost-sensitive SVMs are used to reduce the negative effect of imbalance. This scheme is noted $CS - SVM$ and is used by default in experiments. Results obtained with fine tuning without thresholding (noted FT) and with classical SVMs (noted SVM) are also reported to highlight the usefulness of adapting training schemes to an imbalanced learning context. The two training schemes are run in parallel at the start of the AL process in order to exploit the one which is more accurate. Transfer learning with SVMs is more likely to be useful at the beginning, until the $\mathbb{D}^L$ is sufficiently large for efficient training of deep models. The switch between the two schemes will occur faster if the content of the unlabeled dataset is visually unrelated to the one in the generic model used for the pretrained model. Cross validation using 80 :20 split is performed for each of the two schemes after each iteration. The average accuracy of each scheme is computed to decide which of them should be used starting from the following iteration. The methods which are non-deterministic in the iterative setting, namely random sampling and the proposed method with auxiliary random sampling are repeated with 5 different seeds.

It is common practice in imbalanced learning [23, 86] to evaluate performance over balanced test datasets. This choice is also made here to give equal importance to each class irrespective of the class distribution in the training dataset.

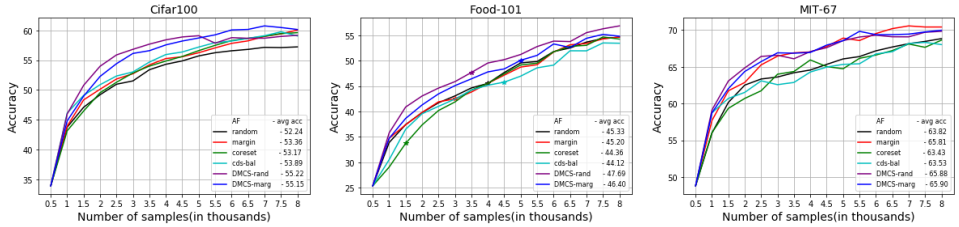


Figure 4.2 – Iterative active learning performance for baselines and for the proposed method using cross-validation between $CS-SVM$ and $FT-th$ training schemes. Results with random ($rand$) and margin ($marg$) based sampling are shown for the remaining budget of each iteration when there are not enough samples associated to minority classes. "*" represents the switching point from $CS-SVM$ to $FT-th$ training scheme. The AL budget is $b = 8000$ and the number of iterations $t = 16$. *Best viewed in color*.

4.3.3 . Datasets

The proposed method and the baselines are evaluated on three imbalanced datasets designed for different visual tasks. As in previous chapter, imbalance is induced in the publicly available CIFAR-100 [123] (object recognition) FOOD-101 [21] (fine-grained food recognition), MIT-67 [182] (indoor scene recognition). The main statistics of the obtained datasets are similar to ones in last chapter shown in Table 3.1.

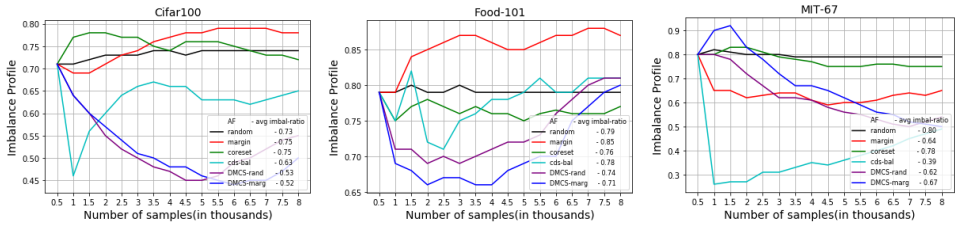


Figure 4.3 – Imbalance profile of labeled datasets for different acquisition functions. $b = 8000$, $t = 16$. *Best viewed in color*.

4.3.4 . Global performance discussion

The results obtained with the baseline methods and with $DMCS$, the diversified version of the proposed AF, are provided in Figure 4.2. A consistent performance gain is obtained with $DMCS-marg$ and $DMCS-rand$ compared to the baselines. This indicates that the proposed method is appropriate for use in iterative AL for imbalanced datasets. Accuracy improvements are obtained for all three datasets. The average accuracy improvement of $DMCS-marg$ compared to the best baselines reaches 2.9, 1.1, 2.1 points for $CIFAR-100$, $FOOD-101$ and $MIT-67$ respectively. The corresponding improvements for $DMCS-rand$ reach 3.0, 2.4 and 2.1 points respectively. The comparison of $DMCS-rand$ and $DMCS-marg$ is globally favorable to the first method and is discussed

in more detail in Subsection 4.3.5. Mirroring global results, $DMCS - rand$ and $DMCS - marg$ have better performance for a large majority of individual iterations on $CIFAR - 100$ and $FOOD - 101$ from Figure 4.2. They are also better than $margin$ sampling for MIT-67 between 1000 and 3000 samples and results become more mixed afterwards as the uncertainty criteria becomes more reliable. This can be attributed to higher performance for MIT-67 dataset as well as the limited number of samples being classified as minority class in later stages. The better behavior of the proposed method is partly explained by its ability to select candidates for labeling whose distribution is globally more balanced, as illustrated in Figure 4.3. The imbalance profiles of the two $DMCS$ versions are clearly better than those of baselines for CIFAR-100 and FOOD-101. They are comparable to those of $margin$ for MIT-67. It is noteworthy that the imbalance profile is not the only factor explaining AF performance. This is clear from the analysis of $cds - bal$ imbalance profile, which is better than that of other methods but is not correlated with a performance gain. The intrinsic quality of the selected samples is also important. The reported results indicate that $DMCS$ is able to provide a more appropriate sampling than the other methods.

The performance of baselines methods is generally close to that of random sampling or even lower. The only exception is $margin$ for MIT-67, which is clearly better than $random$. This result confirms previous reports [15, 198] that random sampling is a competitive baseline in active learning. This is particularly the case for the imbalanced datasets tested here. $coreset$ is comparable to $random$ for $CIFAR - 100$, but is inefficient for $FOOD - 101$ and $MIT - 67$. It also fails to provide any significant improvement in the imbalance profiles for the datasets. It is likely that, for imbalanced datasets, $coreset$ selects outliers that belong to majority classes. $cds - bal$ is useful to tackle cold start in AL and gives best performance in the early stages of the AL process. $cds - bal$ becomes sub-optimal in the later stages when the methods based on the target models become more efficient.

The results show that transfer from a general pre-trained model is preferable at the beginning for all three datasets. Surprisingly, this training scheme remains better than fine-tuning for CIFAR-100 and MIT-67 throughout the entire AL process presented in Figure 4.2. As illustrated, the switch from SVM toward fine-tuning occurs only for $FOOD - 101$. This is intuitive since this dataset was shown to be furthest away from $ILSVRC$ in Chapter 3. The performance of the two schemes is illustrated in detail in Figure 4.6 and further discussed in Subsection 4.3.7. This finding is interesting insofar it is at odds with the usual assumption that fine tuning schemes should be used in iterative active learning [15, 198, 57]. It is also interesting because the transfer learning scheme is much faster since only shallow classifiers need to be trained for each iteration.

4.3.5 . Comparison of random and margin as auxiliary AFs

In Figure 4.2, $DMCS$ results are presented with $random$ and $margin$ as auxiliary AFs if there are not enough samples associated to minority classes. It is

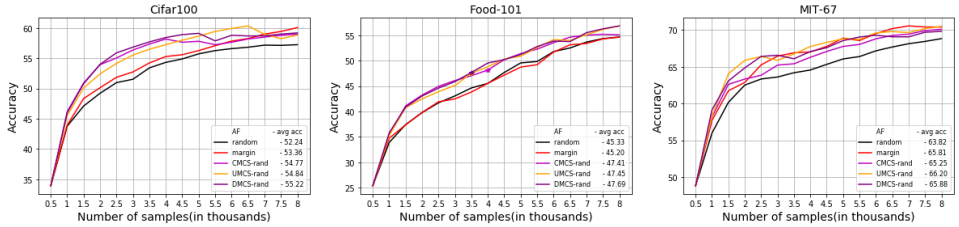


Figure 4.4 – Iterative active learning performance of different versions of the proposed method. Random sampling is used as AF for remaining samples of each iteration if there are not enough samples associated to minority classes. $b = 8000$, $t = 16$. *Best viewed in color*.

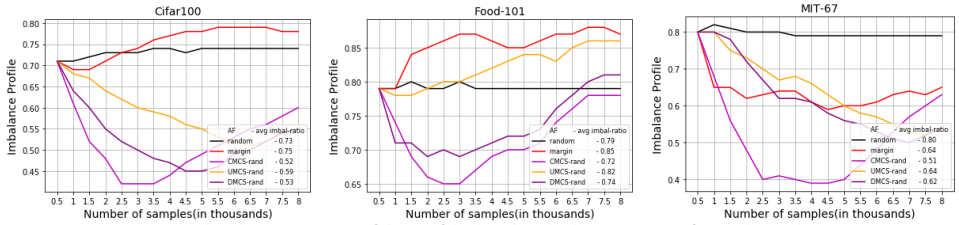


Figure 4.5 – Imbalance profile of labeled datasets for the three versions of the proposed method. Random sampling is used as AF for remaining samples of each iteration if there are not enough samples associated to minority classes. $b = 8000$, $t = 16$. *Best viewed in color*.

somewhat surprising to note that $DMCS - rand$ provides slightly better overall accuracy compared to $DMCS - margin$. This happens even though $margin$ baseline is globally better than $random$ when used alone. $DMCS - rand$ is better for all iterations for the $FOOD - 101$ dataset although the imbalance profile in Figure 4.3 is better for $DMCS - marg$. The difference between the two $DMCS$ variants is very small for $MIT - 67$. Their performance for $CIFAR - 100$ is interesting as $DMCS - rand$ is more effective in up to 5000 samples and $DMCS - marg$ becomes better afterwards. The change of performance is correlated with an inversion of imbalance profiles in Figure 4.3.

We assume that some amount of randomness is effective in the beginning for driving the balancing procedure to focus sampling on minority classes. There, random sampling provides a better overall representation than margin sampling. Later in the AL process, uncertainty estimates are more reliable and imbalance has been mitigated to the extent possible for the given dataset. Then, it becomes preferable to select the remaining samples based on margin sampling.

4.3.6 . Analysis of minority oriented sampling versions

The performance and imbalance profiles for three versions of the proposed method described in Subsection 4.2 are presented in Figures 4.4 and 4.5 respectively. The comparison is done with $random$ as auxiliary AF. All the three versions have a positive impact in mitigating the imbalance with $DMCS$ providing slightly better global performance. This finding indicates that a diversified selection of samples

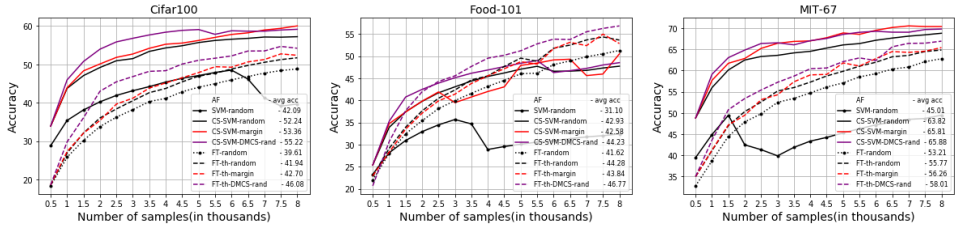


Figure 4.6 – Comparison of classical (SVM, FT) and imbalance-oriented training schemes (CS-SVM, FT, FT-th). $b = 8000$, $t = 16$. *Best viewed in color*.

for minority classes is better than favoring the most certain or uncertain ones. Accuracy is slightly better for $DMCS$ and $CMCS$ for CIFAR-100 in the initial iterations, while $UMCS$ is better later. This is partly explained by the imbalance profile, which increases for the first two versions but not for the third after a certain iteration. For MIT-67, $UMCS$ is most effective since the average performance is higher and the uncertain samples become the ones of interest. This is also the case of CIFAR-100, but later in the AL process.

$CMCS$ is most effective to mitigate the imbalance in the early stages, but leads to most imbalanced dataset by the end of the learning process for CIFAR-100 and MIT-67. It is likely that $CMCS$ learns a limited representation of the minority class, since it focuses on most certain samples. This reduces the model's ability to find samples for the class and also explain the observation that $CMCS$ is outperformed by $DMCS$ and $UMCS$ in later iterations.

The results for FOOD-101 are particularly interesting since the three selection processes lead to different imbalance profiles while the accuracy is quite similar.

4.3.7 . Comparison of training schemes

We illustrate the results obtained by the two training schemes without (SVM, FT) and with ($CS-SVM$, $FT-th$) adaptation for an imbalanced context in Figure 4.6. $CS-SVM$ and $FT-th$ outperform SVM and vanilla FT, their classical counterparts for $random$. The gain is quite significant for $CS-SVM$, validating its use in class imbalanced active learning [50]. Further, we show the effectiveness of the proposed method $DMCS-rand$ over $random$ and the overall strongest baseline $margin$ for both of the two imbalance adapted schemes($CS-SVM$, $FT-th$). This shows the need to explicitly focus on minority classes during the selection process and the effectiveness of the method irrespective of training scheme. Another interesting remark is that, except for classical $SVM-random$, all other transfer learning based schemes outperform fine-tuning schemes for CIFAR-100 and MIT-67. Consequently, both training schemes should be tried at the beginning of the AL process for a new unlabeled dataset. If the distance between a generic dataset and the unlabeled one is not high, transferring features from the first toward the second seems preferable to fine tuning. Otherwise, fine tuning become better at some point during AL and can replace the transfer learning

Dataset	Class	Images	Mean(μ)	Std(σ)	ir
CIFAR-100	100	17168	171.68	126.9	0.74
CIFAR-100-im1	100	19343	193.43	96.36	0.50
CIFAR-100-im2	100	19720	197.20	65.03	0.33

Table 4.1 – Dataset statistics. ir is the imbalance ratio.

scheme, as in case of FOOD-101.

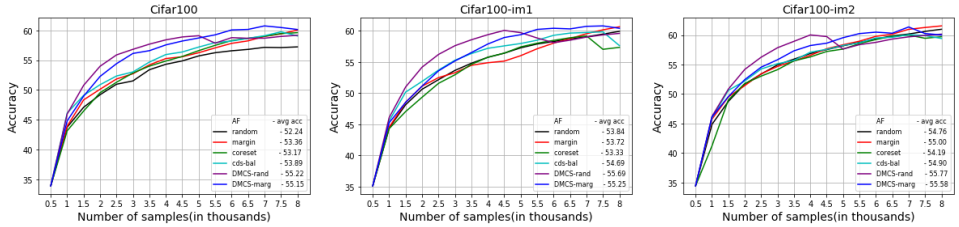


Figure 4.7 – Iterative active learning performance for baselines and for the proposed method using cross-validation between $CS - SVM$ and $FT - th$ training schemes over different imbalance ratios for CIFAR100. Results with random ($rand$) and margin ($marg$) based sampling are shown for the remaining budget of each iteration when there are not enough samples associated to minority classes. The AL budget is $b = 8000$ and the number of iterations $t = 16$. *Best viewed in color*.

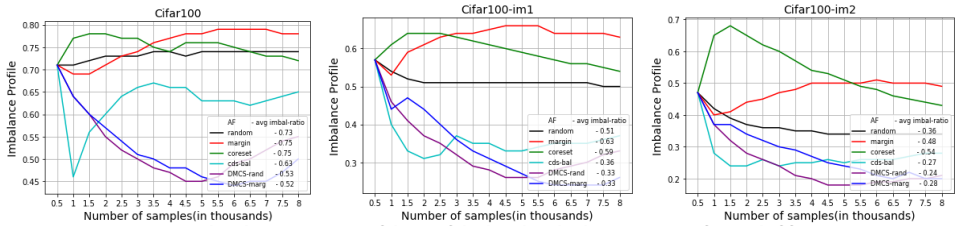


Figure 4.8 – Imbalance profile of labeled datasets for different acquisition functions. $b = 8000$, $t = 16$. *Best viewed in color*.

4.3.8 . Analysis with different dataset imbalance

Experiments were performed with three degree of imbalance with one of the datasets to show the effectiveness of method at different imbalance ratios. We ran experiments with CIFAR-100 and created three imbalanced versions with imbalance ratios of 0.74, 0.50 and 0.33, named CIFAR-100, CIFAR-100-im1 and CIFAR-100-im2 respectively (Table 4.1). The results for CIFAR-100 which has ir-ratio as 0.74 is same to the ones presented above.

The experiments are run in the same budget setting as the main paper with total budget b= 8000 and the total number iteration t set to 16. We observe gains overs the baseline methods for all the three imbalance ratios as shown in Figure 4.7. The imbalance profiles of different methods is provided in Figure 4.8. The main observation is similar over the three imbalance ratios with $DSMC - rand$ being the best method. Further, we also test the three versions of the proposed method

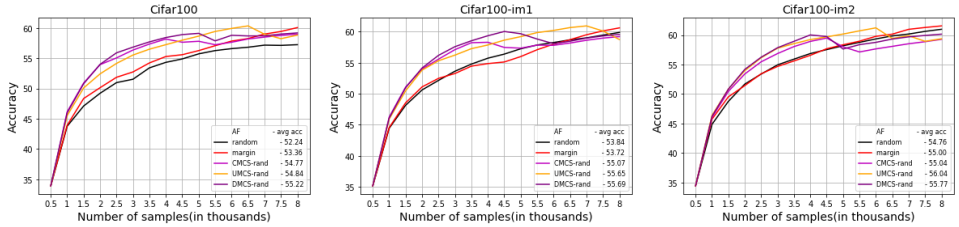


Figure 4.9 – Iterative active learning performance of different versions of the proposed method. Random sampling is used as AF for remaining samples of each iteration if there are not enough samples associated to minority classes. $b = 8000$, $t = 16$. *Best viewed in color Best viewed in color*.

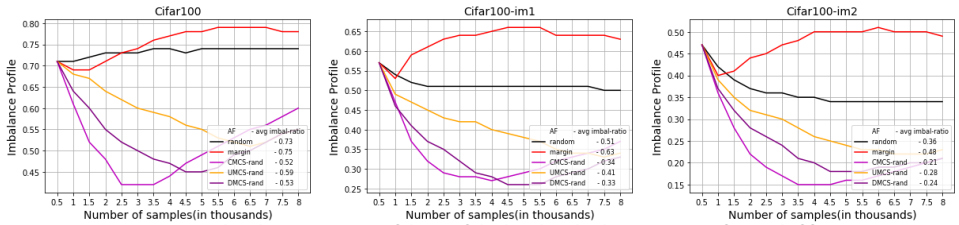


Figure 4.10 – Imbalance profile of labeled datasets for different acquisition functions. $b = 8000$, $t = 16$. *Best viewed in color*.

to select samples from minority classes namely $DSMC - rand$, $CSMC - rand$ and $USMC - rand$ in Figure 4.9. The imbalance profile for the three versions are provided in 4.10. $DSMC - rand$ provides better results over the three imbalance ratios, particularly at earlier stages of iterative process.

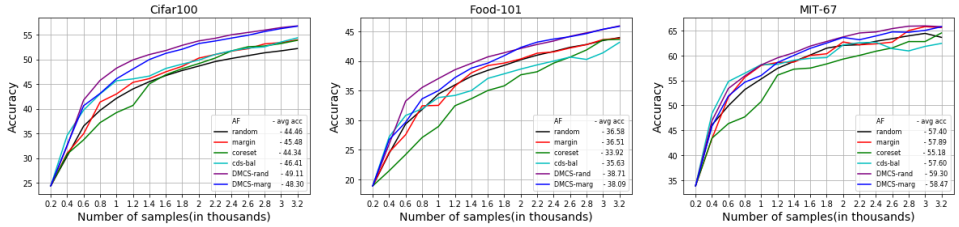


Figure 4.11 – Iterative active learning performance for baselines and for the proposed method using cross-validation between $CS-SVM$ and $FT-th$ training schemes. Results with random ($rand$) and margin ($marg$) based sampling are shown for the remaining budget of each iteration when there are not enough samples associated to minority classes. "*" represents the switching point from $CS-SVM$ to $FT-th$ training scheme. The AL budget is $b = 3200$ and the number of iterations $t = 16$. *Best viewed in color*.

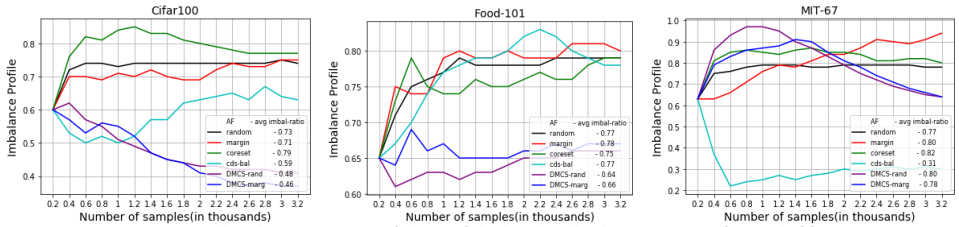


Figure 4.12 – Imbalance profile of labeled datasets for different acquisition functions. $b = 3200$, $t = 16$. *Best viewed in color*.

4.3.9 . Experiments with smaller budget

We addition to testing the methods with a budget of 8000 as presented above, we also test with a smaller budget with $b = 3200$ samples. In this setting 200 samples are added at each iterative step instead of 500. The lower budget setting becomes important in cases where the annotation cost is much higher than the computational cost. The accuracy of methods along with the imbalance profiles are provided in Figure 4.11 and 4.12 respectively. $CS-SVM$ training scheme outperforms FT for all the datasets at lower budgets. Further, the gains become slightly more pronounced in this setting, showing the effectiveness of $CS-SVM$ training scheme for AL task at smaller budgets.

4.4 . Conclusion

We introduce a new acquisition method which is designed for iterative active learning over imbalanced datasets. The method focuses the selection process toward samples which are associated to minority classes in order to reduce the negative effect of imbalance. Evaluation is performed against competitive baselines for active learning, while also applying an effective post-scaling method to tackle affect of imbalance. The proposed methods ensure a performance gain showing that it is important to mitigate the transfer of imbalance during the selection process and techniques . An analysis of its main components facilitates the understanding of their individual contributions. Surprisingly, we find that transfer learning scheme outperforms the fine tuning based scheme usually deployed in AL. We also propose a simple but effective way to test the accuracy of the two schemes after each iteration in order to decide which one should be used later in the AL process.

The results presented here are encouraging and research would be pursued along three axes. First, the proposed method will be tested on larger datasets to understand its behavior for AL at scale. Second, the idea to prioritize minority classes can be extended to balanced dataset to favor classes that are difficult to learn. Finally the effect of the pretrained dataset on transfer learning will be assessed. To do this, ILSVRC can be replaced with a larger dataset, such as the entire ImageNet, for pretraining.

5 - Iterative Active Learning- using asynchronous model predictions

The works in AL use information only from one model i.e. the last learned model, while iterative nature of AL cycle allows to store estimates from previously learned models with minimal additional memory storage. In this chapter, we propose a new measure of informativeness based on the evolution of probability distribution between successive iterative states of AL cycle. Samples for which there is a maximum mismatch in classification between the last two learned models predictions are favored. Further, we expand on diversification approaches introduced in previous chapters to combine the informative and representative objectives of active learning. A diversification step allows to select samples with different class predictions and thus introduces a representativeness component in our approach. The evaluation is done are performed with three balanced datasets : Cifar100, Food-101 and a subset of ImageNet classes which are not part of ILSVRC. We also test our method for imbalanced datasets by creating imbalanced version of Cifar100 and Food-101 and using MIT-Indoor67 which is naturally imbalanced. As in previous chapters, ILSVRC itself is used to create the fixed representation. The results indicate that it outperforms the baselines in most of the evaluated configurations.

The outline of the chapter is as follows : First, we provide the context and motivation of our proposed informative measures in the Section 5.1. Then, we describe the proposed methods $alamp$ and the diversified version $alamp - div$ in Section 5.2, followed the experimental analysis in Section 5.3. Finally, we provide the conclusion in Section 5.4.

5.1 . Motivations

Active learning is generally implemented in an iterative fashion, with a new batch of samples being selected for annotation based on the estimates of the last learned model [15, 57, 199]. In our work, we hypothesise that use of estimates from previous iteration could add useful additional information to the selection criteria.

Our main contribution is a new measure of informativeness which integrates predicted probabilities in successive AL iterations. Samples whose prediction states move from certain to uncertain between two iterations are prioritised. The underlying intuition here is that such samples encode information which is missing from the models and should be integrated into them. The measure is illustrated for a two-class problem in Figure 5.1. This view of informativeness is broader than the one incorporated in current uncertainty-based sampling which only considers the decision boundaries of the current iteration.

The proposed sample AF shares a limitation with existing informativeness-

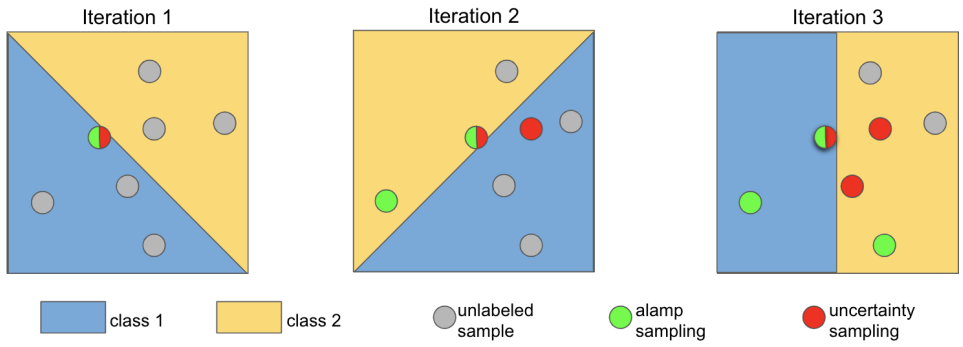


Figure 5.1 – Illustration of $alamp$, the proposed method, for a two class problem. Blue and yellow regions are the class boundaries learned by classifier. One sample is selected at each iteration. Uncertainty sampling selects the sample (red) which is closest to the class boundary. At Iteration 1, $alamp$ selects the same sample as sampling, since it has is no access to previous model. At Iteration 2, $alamp$ selects the sample which gives most certain prediction (i.e. is furthest away from decision boundary) at Iteration 1 and gives the most uncertain prediction (i.e. is closest to decision boundary) at Iteration 2. Similarly at Iteration 3, $alamp$ selects the sample with maximum shift from certainty to uncertainty

based functions in that it could suffer from a lack of representativeness [199]. As discussed earlier, informativeness and representativeness are not easy to optimize jointly but, since they convey complementary cues, their combination could lead to better AL selection and is studied in recent works such as [7, 27]. Our second contribution is to add a representativeness dimension in the proposed acquisition function. Representativeness is modeled via the use of a diversification procedure. The proposed informativeness measure prioritizes samples with high certainty at the previous iteration. The diversification procedure exploits this fact to select samples with different class predictions in the previous iteration. This leads to selection of samples from different uncertain regions distributed across the classification space provided by the model classifier.

In the previous chapters, we explored training of a Support Vector Machine (SVM) classifier over fixed representation as an alternate to the dominant fine tuning scheme used in recent works [15, 57, 59]. This type of approach, which instantiates transfer learning, has been shown to be effective in previous chapters on active learning over imbalanced datasets. Interestingly, the two contributions are well-suited for transfer learning based approach where shallow classifiers are learned over fixed representations. The constant nature of representation helps the proposed measure to effectively evaluate the distance of the samples to the classifier boundary in the preceding and the current iterations. This is important insofar fixed representations can be exploited from the very beginning for the AL

task where the classical fine tuning estimates are shown to be unstable [59].

5.2 . Proposed method

The proposed method ascertains the informativeness of samples by taking into account the change in their probability distribution in successive iterations. The strategy prioritizes samples which were predicted with high certainty in the previous iteration but which gives uncertain prediction in the current model. We derive an analogy to a student who gave confident response to a question, but becomes uncertain after learning some more information. Knowing the true answer should benefit the student and provide relevant missing information. The strategy is well-suited for the low budget setting where the batch size is generally small. The update of model with large number of samples could make the precedent model less relevant. Focusing on the unlabeled samples on which the model becomes uncertain in its predictions adds a novel component of uncertainty in the selection process. In cases where the sample was correctly predicted, selecting these samples allows the model to focus on samples that it is most likely to forget. Alternatively, even if a sample was predicted incorrectly, the measure allows to select more difficult samples which can improve the generalization ability of the model. Hence, knowing the labels of these samples should be informative.

5.2.1 . alamp : active learning with asynchronous model predictions

We present here, our formulation of $alamp$ that allows to select the samples with the maximum shift from certainty to uncertainty between the two iterations. Note that any of the uncertainty measures can be used in $alamp$. The definition of the method which exploits margin sampling as basic acquisition function is :

$$alamp_k(x, \theta_{k-1}, \theta_k) = \frac{marg_{k-1}(x, \theta_{k-1}) - marg_k(x, \theta_k)}{marg_{k-1}(x, \theta_{k-1}) + marg_k(x, \theta_k)} \tag{5.1}$$

with $marg_k(x, \theta_k$, the margin function as in equation 5.2 gives the uncertainty score for samples x from $\mathbb{D}_k^U$ at iteration k.

$$marg_k(x, \theta_k) = p_k(\theta_k, \hat{y_1}|x) - p_k(\theta_k, \hat{y_2}|x) \tag{5.2}$$

where $\hat{y_1}, \hat{y_2}$ are the top-2 predicted classes for test sample x at iteration k.

The score takes a normalized min-max view of uncertainty allowing to select samples with maximum change (certainty to uncertainty) between the iteration. The sample with higher score is selected. The numerator gives the difference in the certainty between the previous and current iteration. The denominator normalizes the certainties to ensure that for samples which same absolute difference in certainty(numerator) one with lower sum of certainties is selected. This allows the score to select samples with maximum relative shift towards uncertainty.

We consider $\mathrm{Alamp}_{\mathrm{sort}}(\mathbb{D}_k^U)$ a permutation of the set $\mathbb{D}_k^U$ by ordering its element by decreasing value of $alamp$. $\mathbb{D}_{k+1}^L$ is obtained by the union of $\mathbb{D}_k^L$ with the first $\frac{b}{t}$ samples of $\mathrm{Alamp}_{\mathrm{sort}}(\mathbb{D}_k^U)$.

The method has low supplementary memory requirements since it only stores the probability distributions at each iterative step. At the first iterative step, we have access to only the probability estimate $\mathcal{P}_0$ from the initial model $\mathcal{M}_0$, which is trained over the initial randomly selected dataset $\mathbb{D}_0^L$, thus the selection is based on the uncertainty criteria in the first iterative step.

5.2.2 . alamp-div

$alamp$ inherits the limitations of the uncertainty-based method used in its definition in terms of sample representativity. We introduce a variant of the method, named $alamp-div$ which selects informative samples from different regions of the classification space. $alamp-div$ is described in Algorithm 2 is similar to the one presented in Section 3.3. Instead of using the source model to assign samples to classes, as in single stage AL setting, samples are assigned to classes predicted in the previous iteration. $alamp$ prioritises the sample with high certainty in the previous iteration, while being uncertain in the current iteration. Thus, The selection process is driven toward selecting the same number of samples from each pseudo class so to aim for representativeness and balance across classes. This enables the selection of informative samples across a diverse set of classifier boundaries in a multi class problem.

5.3 . Experiments

We first describe the experimental setup for the transfer and fine-tuning training schemes that are tested. Then, we describe the evaluation datasets. Finally, we present the results and their analysis.

5.3.1 . Setup

The experimental setup is designed to focus on the small annotation budgets, which is most challenging for AL. In our experiment, we use 200 samples for the transfer and fine-tuning initial AL budget. The AL process is then run for 15 iterations with 200 samples selected at each iteration. The total budget at the end of the process includes 3200 samples. Further, we test the performance of proposed methods at a higher budget setting of 8000 samples with 500 samples selected for each of the 15 iteration.

We experiment with two training schemes. For comparability, a ResNet-18 architecture [88] is used as backbone of both of the training schemes tested here. The first scheme, noted FT, is based on fine-tuning and mirrors the dominant approach in existing deep AL works [57, 198]. We fine-tune the pre-trained model for 80 epochs. All the parameters are optimised using stochastic gradient descent with Nesterov momentum of 0.9. The initial learning rate is 0.01 and is reduced by

Algorithm 2 Diversification algorithm

U : a list of unlabeled samples sorted according to $alamp$
k : current iterative step
top : a dictionary which assigns top class prediction in the iteration $k-1$ to all samples in U
b : budget of samples to be selected
procedure div(U, top, b)
    Build L : list of samples selected from U of length b
    while len(L)$\leq b$ **do**
        seenclasses = empty list : reinitialize memory of classes
        for each item i in U **do**
            $topclass$ = top[i] :predicted class at iteration $k-1$ for sample $U[i]$
            if $topclass$ not in $seenclasses$ **then**
                if i not in L **then**
                    add sample i in L
                    add $topclass$ in $seenclasses$
                end if
            end if
        end for
    end while
    $L = L[0:b]$
    return L
end procedure

Dataset	Class	Train images/class	Test images/class
Cifar100	100	500	100
Food-101	101	750	250
IMN-100	100	1000	200

Table 5.1 – Dataset statistics.

a factor of 10 when the train error rate plateaus for 10 epochs. We use a weight decay parameter of 0.001. The models are trained using the Pytorch framework. Thresholding [23] which is shown to be effective to mitigate imbalance for deep models is used in experiments with imbalanced datasets. The experiments are repeated for 5 runs and the average performance is reported.

FT can be suboptimal when the budget is small enough to optimize the large number of parameters of the DNN. An alternate to FT is to learn a SVM classifier over the features of the pre-trained model. Transfer learning scheme, noted as SVM here, has been proven to be beneficial when the number of annotated AL samples is limited [2]. In our work, we exploit the features of a pre-trained model on $ILSVRC$ [193] dataset. The scikit-learn implementation of SVC classifier is used with standard default parameters. The cost sensitive SVM implementation is used for imbalanced datasets. The regularization parameter is selected using a cross-validation on the training data. While sub-optimal, the use of training data for validation necessary because of data scarcity specific to AL. The SVM training scheme is deterministic once the initial subset is selected.

5.3.2 . Datasets

The acquisition function and the training schemes are tested on three publicly-available image classification datasets $Cifar100$, $Food-101$ and $IMN-100$. The three datasets are balanced and their main statistics are provided in Table 5.1. In addition, we test our methods on imbalanced versions of datasets $Cifar100$, $Food-101$ and $MIT-67$ [182]. The imbalance induced is similar as in previous chapters.

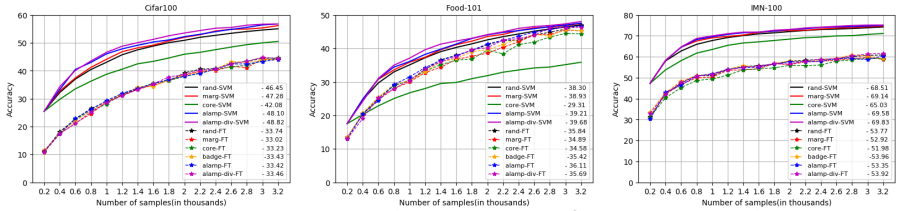


Figure 5.2 – Iterative active learning performance with SVM and FT training schemes at each of 15 iterative steps for **balanced** datasets Cifar100, Food-101 and IMN-100 with initial budget of 200 and total budget of 3200. 200 samples added at each iteration *Best viewed in color*.

5.3.3 . Analysis of results

The results obtained for the three balanced datasets using the baselines and the proposed methods are presented in Figure 5.2. Note that we provide both the detailed evolution of accuracy across AL iterations and the averaged performance of each method. Globally, $alamp$ and $alamp - div$ provide the best performance across the three datasets for SVM training scheme. $random$ acts as a strong baseline, especially for FT training scheme, where none of the methods that are tested can outperform $random$. Further the performance of SVM scheme is clearly higher than that of the usual FT scheme. This is an interesting result which is analyzed in detail in Subsection 5.3.4.

Here we discuss the performance of different AFs in the SVM training scheme. The average accuracy gain for the entire AL cycle is 2.4 , 1.3 and 1.3 points for $alamp$ compared $rand$ for balanced versions of $Cifar100$, $Food-101$ and $IMN-100$ respectively. More interesting from a practical perspective, the number of samples required for achieving 50 percentage of accuracy for $Cifar100$ is 1800 with $random$ or $marg$, 1600 for $alamp$ and 1400 for $alamp - div$. Similarly for $Food - 101$, 40 percent accuracy is reached with around 1600 samples for $random$, 1400 samples with $alamp$ and $marg$ and 1200 samples for $alamp-div$. $alamp-div$ avoids the annotation of 400 extra samples as compared to $random$ to achieve 50 percent accuracy for $Cifar100$ and 40 percent accuracy for $Food-101$. The performance gain is more limited for $IMN - 100$. Overall accuracy for this dataset is already quite high with SVM training scheme. This is an expected result, since $IMN - 100$ is closest to the $ILSVRC$ dataset used to train the source model. $alamp - div$ and $alamp$ are still the best methods with 70 percent accuracy attained with 1000 samples with $alamp$ and $alamp - div$, while $rand$ and $marg$ require 1200 samples.

$marg$ outperforms $random$ in the SVM training scheme, showing that SVM classifier provides reliable uncertainty estimates even at low budgets. In our experiments, $marg$ becomes competitive to $alamp$ at large budgets when the uncertainty estimates become stronger. This is explained by the fact that, as the accuracy of the model increases, uncertainty measures becomes more important to find the missing information. This is the case of $IMN-100$, which has the highest overall accuracy among the three dataset tested. The accuracy of $IMN - 100$ is around 50 percent at the start of AL cycle and $marg$ is more competitive for $IMN-100$ than for the other two datasets.

$core$ has suboptimal performance in the SVM training scheme for all datasets. This is particularly the case of $Food - 101$, which is most different from the $ILSVRC$ dataset used as feature extractor. It gives comparable performance to other AFs in the FT training scheme where the feature extractor is updated along with the classifier. $badge$ is suited only for FT training scheme as it requires the gradients on the features. In SVM training scheme, the features are fixed and hence it is not possible to test $badge$. In the FT training scheme, $badge$ also fails

to provide any significant improvement over $random$.

5.3.4 . Analysis of training schemes

A initial subset is needed to start the iterative AL process. It has two main impacts on the FT scheme as seen in Figure 5.2. First, at low budgets the fine-tuned model fails to provide strong probability estimates for the acquisition function . This is evident from the results where none of the tested AFs is able to conclusively outperform $random$ sampling. It is also the case for $badge$ which has shown improvement over $random$ in [7]. The key difference is the lower budget setting studied in our work.

Second, the comparative analysis of the two training schemes shows that FT is largely outperformed by the transfer learning strategy for low AL budgets. The performance of FT scheme starts at 11.16, 13.19 and 32.65 percentage points for balanced $Cifar100$, $Food-10$ and $IMN-100$ respectively. The corresponding accuracy with SVM is significantly higher, with 25.5, 17.54 and 47.29 percentage points respectively. This is somewhat intuitive since deep model can be accurately trained only if a relatively large amount of data is available. FT lags behind even at the end of the AL process for $CIFAR100$ and $IMN-100$ but becomes competitive for $Food-101$. This last result is explained by the lower similarity between $Food-101$ and $ILSVRC$ compared to the other datasets. The efficiency of transfer learning is lower for this dataset, but still much better than fine-tuning in early phase AL.

The comparison of FT and SVM schemes has practical implications for AL. The training process is much quicker with SVM since it only requires an update of the shallow classifiers. Further, a cross-validation step can be envisaged to switch from SVM to FT training when FT outperforms SVM training scheme. As suggested by $Food-101$ results, this happens once there are enough samples for a competitive training of deep models.

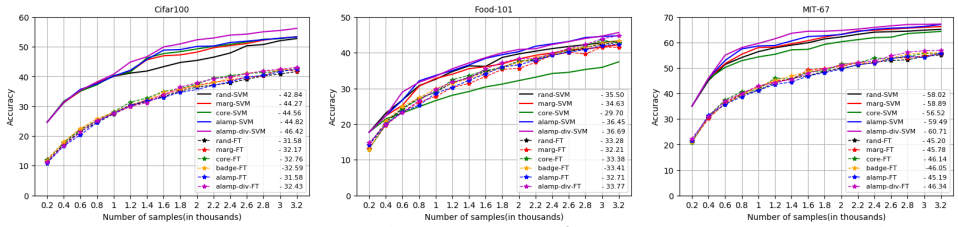


Figure 5.3 – Iterative active learning performance accuracy with SVM and FT training schemes at each of 15 iterative steps for **imbalanced** datasets Cifar100, Food-101 and MIT-67 with initial budget of 200 and total budget of 3200. 200 samples added at each iteration *Best viewed in color*.

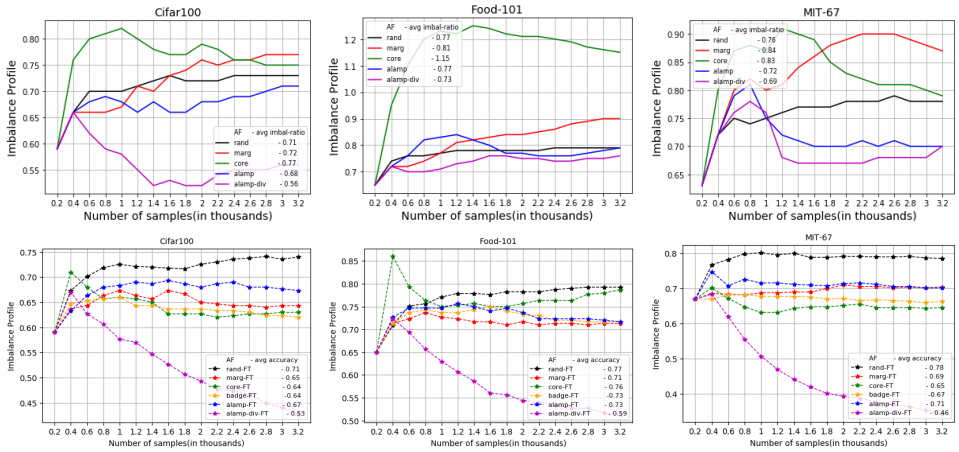


Figure 5.4 – Imbalance Profiles with SVM(top) and FT(bottom) training schemes at each of 15 iterative steps for **imbalanced** datasets Cifar100, Food-101 and MIT-67 with initial budget of 200 and total budget of 3200. 200 samples added at each iteration *Best viewed in color*.

5.3.5 . Impact on imbalanced datasets

The performance of the methods on imbalanced dataset for SVM scheme of annotated subset is presented in Figure 5.3. Both $alamp$ and $alamp-div$ provide improvement over the baselines methods. $alamp$ provides average gain of 1.99, 0.9 and 1.49 points for $Cifar100$, $Food-101$ and $MIT-67$ respectively. The diversification component is particularly more effective for imbalanced datasets with gains of 3.59, 1.14 and 2.71 points respectively. For example, 50 percent performance on $Cifar100$, is reached with 1600 samples for $alamp-div$, while it takes atleast 2000 samples for any other best method. A possible explanation can be found in the imbalance profile of selected subsets (Figure 5.4). The imbalance profiles show the effectiveness of the methods $alamp$ and $alamp-div$ to mitigate the imbalance from being propagated to labeled subset. The results are reported after the use of effective techniques from imbalanced learning. The improvements with the proposed methods also shows the importance of tackling imbalance at the time of sample selection for imbalanced datasets.

5.3.6 . Impact of diversification

The diversification procedure is effective for both balanced and imbalanced dataset, where $alamp-div$ improves results over $alamp$. The key reason for its effectiveness is that $alamp$ prioritizes samples with high certainty in the previous iteration. Even though the class prediction changes after the update of the model, samples having different class prediction with high confidence at the previous iteration are likely belong to different regions of representation space.

We also test the diversification procedure for standard acquisition functions $rand$ and $marg$ on $Cifar100$. $core$ is not considered here as it already selects representative samples and also is not competitive with other AFs. The pseudo class

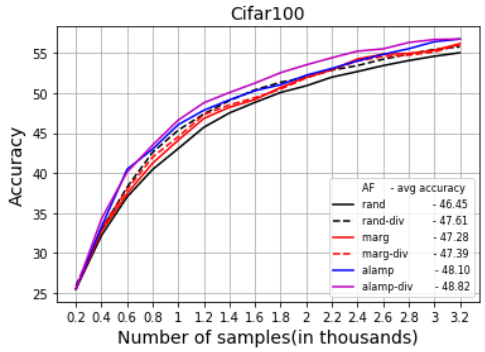


Figure 5.5 – Iterative active learning performance for diversification applied to $rand$ and $marg$ with SVM training schemes at each of 15 iterative steps for Cifar100 with initial budget of 200 and total budget of 3200. 200 samples added at each iteration.

for $rand$ and $marg$ is assigned using the current class prediction. The diversification results are presented in Figure 5.5 , with $rand-div$ and $marg-div$ with diversified version of $rand$ and $marg$ respectively.

$alamp-div$ still provides the best performance, but interestingly $rand-div$ outperforms $rand$. The gain for $rand-div$ is particularly higher at the start of the iterative cycle, where representative sampling is shown to be more important. $marg-div$ has very little effect compared to $marg$. This is expected since $marg$ sorts the samples in terms of uncertainty. Thus, the class predictions are not reliable and the diversification procedure becomes ineffective.

5.3.7 . Analysis with larger batch size

The results obtained for the three test datasets for a higher budget setting is shown in Figure 5.6. The initial budget is set to 500 samples, with AL cycle run for 15 iterations adding 500 samples at each iteration. The final budget is 8000 samples. The results for the two training schemes are presented separately.

The proposed methods $alamp$ and $alamp-div$ provide the best overall results for the three datasets. The gain is most interesting for $Cifar100$ and $Food-101$. $IMN-100$ dataset performs the best at higher budgets and performance of proposed method is close to uncertainty sampling. Similarly, we also test in higher budget setting for the three imbalanced datasets. The initial budget is 500 and 500 samples are added for 15 iteration, for final budget to be 8000 samples. The accuracy and imbalance profiles are presented in Figure 5.7 and Figure 5.8 respectively. The proposed methods $alamp$ and $alamp-div$ provide the best overall results for the three datasets. $alamp-div$ gives the best results as it able to mitigate the imbalance to the labeled set.

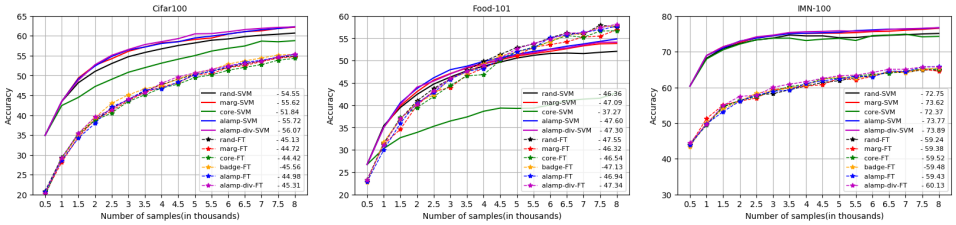


Figure 5.6 – Iterative active learning performance with SVM and FT training schemes at each of 15 iterative steps for **balanced** datasets Cifar100, Food-101 and IMN-100 with initial budget of 500 and total budget of 8000. 500 samples added at each iteration *Best viewed in color*.

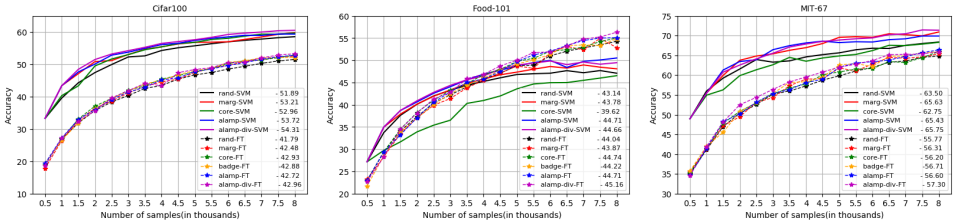


Figure 5.7 – Iterative active learning performance accuracy with SVM and FT training schemes at each of 15 iterative steps for **imbalanced** datasets Cifar100, Food-101 and MIT-67 with initial budget of 500 and total budget of 8000. 500 samples added at each iteration *Best viewed in color*.

5.4 . Conclusion

The main contribution of this work is the introduction of two new acquisition functions. $alamp$ and $alamp - div$ capture the dynamic nature of probability estimates of iterative AL models. They outperform competitive baselines over both balanced and imbalanced image classification datasets. A diversification component is introduced to combine the objectives of informativeness and representativeness. We tested the diversification procedure for random sampling, margin sampling and our proposed method. The diversified version of $alamp$ is particularly effective compared to the other two sampling methods as $alamp$ provides strong pseudo class predictions using the certainty measure from the previous iteration.

We also tested the diversification procedure for random sampling and margin sampling. The diversified version of random sampling gives some improvement and performs even better than margin sampling. The diversified version of margin gives slight improvement at the beginning of the iterative cycle where diversification is more important.

The result of the proposed informative measure and the diversification procedure is inconclusive for FT training scheme. This could be a result of evolving representation space with fine-tuning of model. In the future, we plan to explore ways to implement the proposed informative measure for fine-tuning scheme using different snapshots during the fine-tuning process. Further, efficacy of the proposed

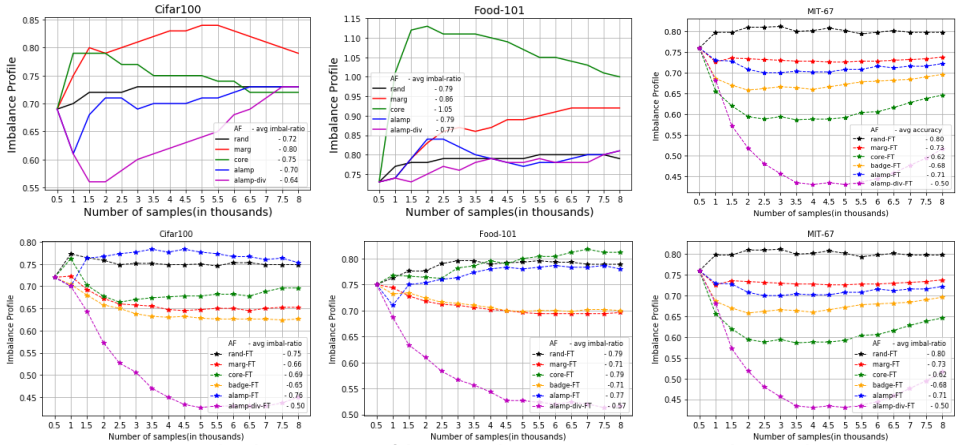


Figure 5.8 – Imbalance Profiles with SVM(top) and FT(bottom) training schemes at each of 15 iterative steps for **imbalanced** datasets Cifar100, Food-101 and MIT-67 with initial budget of 500 and total budget of 8000. 500 samples added at each iteration *Best viewed in color*.

methods with different batch sizes can be studied. The selection of AL strategy can be explored based on the imbalance ratio present in the selected labeled set. In case of higher imbalance and applications where minority classes are of importance, the method presented in Chapter 3 is becomes suitable since it prioritizes the selection of minority classes. While, for balanced datasets $alamp$ is a better option since it allows to combine the informative and representative objectives.

6 - Incremental learning over imbalanced dataset

In this chapter, we focus on the intersection of incremental learning and class imbalanced learning. Each problem is well studied separately but, to our knowledge, they were not tackled together in the context of deep learning. A joint study is needed to cope with dynamic and imbalanced datasets. The focus is on challenges related to deep architecture complexity and to scalability. These properties are of utmost importance in applications such as visual content analysis. The visual corpora to be analyzed evolve quickly and there is a need for updating the underlying classification models accordingly. Here we advocate that class incremental learning with a bounded memory actually boils down to a form of imbalanced learning problem. New data most often corresponds to majority classes, while old data corresponds to minority classes since images of old classes need to be fit in the bounded memory that is allocated to them. In our work, we study various calibration methods and propose two novel ones to reduce the bias between majority and minority classes. Further, we also evaluate the model calibration with each of the calibration method.

We outline the structure of the chapter. Section 6.1 introduces the problem of incremental learning in the presence of dataset imbalance. Section 6.2 formalizes the incremental learning with imbalanced datasets problem. Section 6.3 discusses calibration as an effective way to counter dataset imbalance and introduces the different calibration methods tested. Section 6.4 compares the calibration methods to three strong incremental learning baselines and proposes an analysis of results in terms of accuracy (Subsection 6.4.4) and the ability to provide calibrated predictions (Subsection 6.4.5). Finally, Section 6.5 presents the conclusions and perspectives related to the proposed contribution.

6.1 . Introduction

Large scale image repositories are highly dynamic, with content being added and/or removed at a fast pace. However, content analysis is currently done with algorithms built to learn from static information. This is notably the case for deep learning models which are trained on fixed datasets. When model updates are required, the entire training corpus is reused for learning, making the process cumbersome. To make image content analysis more dynamic and thus more adapted to dynamic corpora, incremental [188] or lifelong [4] learning processes need to be implemented.

Recent research in incremental learning use deep learning as backbone and most of them focus on class incremental learning, a setting in which data are

completely labeled. The main challenge in incremental learning is due to a restricted or impossible access to old data. As discussed in section 2.4 if no memory of the past is allowed, the deep architecture grows in time to accommodate new classes [4, 195, 233]. If a bounded memory is allowed, the architecture is fixed and an adapted fine tuning is applied to learn incrementally [25, 109, 188]. Fine tuning based incremental learning methods are akin to an imbalanced learning problem due to the availability of a fixed size memory for past classes. Imbalance worsens as more classes are added incrementally since exemplars need to be fit in memory for all past classes.

In our work, we study the calibration methods whose objective is to reduce the prediction bias between majority and minority classes. We compare the following calibration methods : (1) isotonic regression [244] and Platt scaling [180] which leverage initial scores to improve final predictions, (2) thresholding applied to the initial class probabilities in order to increase the predictions of rare classes [23], (3) nearest-exemplar-mean classifier [188] and balanced fine tuning [25] which were recently introduced as post-processing steps to reduce the effect of data imbalance in deep incremental learning and (4) two proposed methods which group classes in batches either as new vs. old or by image counts and then exploit the mean classification scores per batch for calibration.

While the focus is on methods which increase the performance on the test dataset, we also evaluate the intrinsic effect on model calibration. Deep learning models have been shown to provide over-confident predictions that we do not match its accuracy[73]. Well-calibrated models have the confidence levels aligned to the model accuracy and thus give valuable information of how likely the model is to be correct or incorrect.

Evaluation is done with three large scale datasets designed for object, face and landmark recognition. Existing methods reduce the effect of imbalance via the use of class exemplars [188] or balanced fine tuning [25]. We include these methods in our study of calibration methods. We also include more recent methods such as BiC [238] and $LUCIR$ [97] as baselines to evaluate their performance on imbalanced datasets.

The main findings are :

1. the obtained results support the usefulness of a majority of post-processing methods for the reduction of bias toward majority classes
2. when a bounded memory is available, the use of vanilla fine tuning followed by calibration is preferable to the widely used distillation loss [25, 109, 188, 97, 238].
3. thresholding based calibration is most effective in providing overall improvement in accuracy, though it has detrimental affect on model calibration. The proposed methods provide consistent improvement in both model accuracy and model calibration.

6.2 . Problem formalization

We consider $\mathcal{D}_N$ a labeled dataset $X_y, y_t \in \mathcal{X} \times \mathcal{Y}$ for $t = 1, 2, ..T$ i.i.d realizations of random variables $\mathcal{X}, \mathcal{Y} \sim \mathbb{P}$, where $\mathbb{P}$ is the data distribution, $\mathcal{X}$ is the instance space and $\mathcal{Y}$ is the set of N class labels $\{y_1, ..., y_N\}$. We denote $X_i = \{x_i^1, x_i^2, ..., x_i^{n_i}\}$ the set of n_i instances for the class y_i in the dataset $\mathcal{D}_N$. In a supervised classification problem, the objective is to learn a model $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Y}$ that maps an instance x to a label vector $\hat{Y}$. By the following, we will denote $\hat{Y}$ as a set of the class prediction with $\hat{y_i}$ the prediction score for class y_i.

In an incremental learning setting, at each incremental state k, a set of P_k new classes is added to the previous dataset with, for each new class j, a set of n_j instances. The objective is thus to use $\mathcal{M}_{k-1}$, the model learned at the previous step, as input for an updated model $\mathcal{M}_k$ which classifies $N_k = P_1 + P_2 + ... + P_k$ classes. Here, N_k is the total number of classes that have been observed from the beginning. $\mathcal{M}_k$ is trained using a dataset $\mathcal{D}_{N_k}$ composed of all the instances of the P_k new classes and only a restricted set of the instances of the N_{k-1} old ones. In particular, we assume a bounded memory B is available for the instances of the old classes in each incremental state. As a consequence, due to this limited memory size, $\mathcal{D}_{N_k}$ is by nature imbalanced and imbalance grows at each incremental state.

We consider deep neural models $\mathcal{M}_k$ which include two main components. The first is a feature extractor $\mathcal{F}_k : \mathcal{X}_{N_k} \rightarrow \mathbb{R}^d$, with d the size of the feature vector f. The second is a classifier $\mathcal{C}_k : \mathbb{R}^d \rightarrow \mathcal{Y}_{N_k}$ which outputs the classification scores $\hat{y_i}$ for the N_k learned classes. The classification scores can then be converted to probability estimates $\hat{p_i}$ to ascertain the confidence of the model. Depending on the calibration method used, $\mathcal{F}_k$ and $\mathcal{C}_k$ are either integrated in a single deep model or separated to tackle the bias while learning incrementally over imbalanced datasets.

6.3 . Calibration methods

Dealing with imbalance is important as the number of training samples per class often varies in real-life applications. As a consequence, majority classes have better representations and are favored over minority ones. The application of calibration methods is an effective way to counter the effect of imbalance [23]. Put simply, calibration attempts to boost predictions for minority classes in order to compensate for their weaker representation in the deep model. We study different calibration methods proposed either in imbalanced or incremental learning literature. Fine tuning algorithms for incremental learning update the model $\mathcal{M}_{k-1}$ at an incremental state k with training examples from new classes P_k and a bounded exemplar set from past classes N_{k-1}. If the initial dataset is balanced, we assume that each class is represented by S images. The bounded memory thus generates a binary imbalance with old classes being represented by $\frac{B}{N_{k-1}}$ and new classes by S images. We term this imbalance as incremental imbalance as it arises as a consequence of learning incrementally with a bounded memory. In our context, a

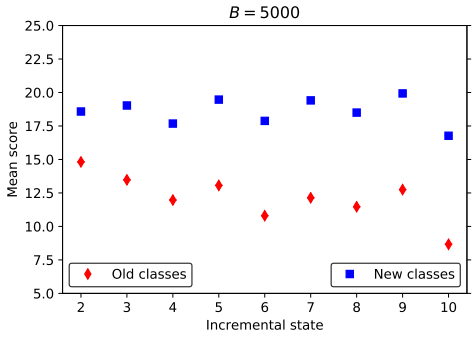


Figure 6.1 – Prediction scores for old and new classes using vanilla fine tuning for the $ILSVRC$ dataset. Training is done with vanilla fine tuning, $B = 5000$ bounded memory for old classes and $soft$ imbalance configuration as defined in Section 6.4. The first, non-incremental, state is not represented.

dataset imbalance due to the variable class image counts is added to the imbalance generated via incremental learning. The imbalance profile is not binary anymore since new classes are represented by a variable number of images and the proposed calibration methods should take this into account.

When the model $\mathcal{M}_k$ is trained with a dataset affected by both incremental and dataset imbalance, it learns a feature extractor which is biased toward majority classes and performance is sub-optimal. On average, the biased classifier associates higher scores to images from majority class than minority classes. Figure 6.1 illustrates this bias using the $ILSVRC$ dataset. The mean score of a class is computed using the predictions obtained for its samples from the training dataset. Then, we aggregate the average over the old and new classes to estimate the mean score of old and new classes. We note that the mean scores of old classes are consistently lower than those of old classes for all incremental states. Moreover, the difference tends to grow from left to right since the imbalance is higher in later incremental states.

We focus on fixed deep architectures and, in this case, the bias induced by imbalance needs to be reduced without increasing the complexity of the feature extractor $\mathcal{F}_k$ for the deep model $\mathcal{M}_k$. **Consequently, calibration methods act as an adaptation of $\mathcal{C}_k$, the classification layer of $\mathcal{M}_k$, with the aim of reducing the bias toward majority classes.**

We first present calibration methods which cover the main approaches from literature. Then we introduce two simple methods which leverage : (1) the prediction score means for old and new classes and (2) the distribution of the number of images per class.

6.3.1 . Isotonic regression calibration (iso)

Isotonic regression [244] transforms the initial classifier predictions into a discrete set of calibrated scores. Since the number of available images is reduced in incremental learning, isotonic regression exploits the training set $\mathcal{D}_{N_k}$. The calibration is performed individually for each class and exploits the overlap between positive and negative examples for each class. A discrete set of scores $R = \{0, ..., p_l, ..., 1\}$ is created where each discrete value represents a range of initial prediction scores. p_l will be assigned to all initial predictions between two consecutive prediction boundaries $\hat{y_i}^{l-1}$ and $\hat{y_i}^{l}$ determined via isotonic regression, with $\hat{y_i}^{l-1} < \hat{y_i}^{l}$. The isotonic calibrated version of $\hat{y_i}$ is :

$$\hat{y_i}^{iso} = p_l, \text{ if } \hat{y_i}^{l-1} \leq \hat{y_i} < \hat{y_i}^{l} \tag{6.1}$$

Isotonic regression has the advantage of being non-parametric but requires a large number of positive samples per class to discretize probabilities in an efficient manner. The method is applied as a post-processing step after $\mathcal{C}_k$. The class predicted after calibration is the $argmax$ value of all calibrated class predictions given by Eq. 6.1.

6.3.2 . Platt calibration (pl)

Platt scaling [180] fits a logistic regression over the initial scores in order to reduce miscalibration. We write the calibrated score as :

$$\hat{y_i}^{pl} = \frac{1}{1 + \exp(A\hat{y_i} + B)} \tag{6.2}$$

where A and B are two parameters which need to be learned and $\hat{y_i}$ is the initial prediction of the i^{th} class. The incremental training set $\mathcal{D}_{N_k}$ is used to determine A and B by optimizing a max likelihood method. There is evidence that isotonic regression outperforms Platt scaling if enough examples per class are available [165]. However, this finding was not tested for imbalanced datasets which may include a lot of minority classes, as it is the case here. The class predicted after calibration is the $argmax$ value obtained by applying Eq. 6.2 to the initial scores predicted by $\mathcal{C}_k$.

6.3.3 . Thresholding based calibration (th)

Thresholding [23] adjusts the prediction scores of a multi-class classifier by dividing the output of a class in $\mathcal{C}_k$ by its estimated prior probability. The calibrated score of is written as :

$$\hat{y_i}^{th} = \frac{\hat{y_i}}{\frac{n_i}{\sum_{l=1}^{N_k} n_l}} \tag{6.3}$$

where n_i is the number of images for the i^{th} class and $\sum_{l=1}^{N_k} n_l$ is the total number of images in the training dataset $\mathcal{D}_{N_k}$. As we mentioned, a recent study of imbalanced learning for deep learning models showed that thresholding is highly

efficient [23]. Its usefulness is theoretically supported by the fact that the outputs of a neural network correspond to Bayesian a posteriori probabilities [190]. The class predicted after calibration is the $argmax$ value obtained over all predictions obtained with Eq. 6.3.

6.3.4 . Nearest-mean-of-exemplars calibration (nem)

The authors of $iCaRL$ [188] proposed nearest-mean-of-exemplars, an adaptation of nearest class-mean classifier [155], to counter the inherent imbalance in incremental learning. The calibrated score of the i^{th} class is written as :

$$\hat{y_i}^{nem} = ||f(x) - \mu_i|| \tag{6.4}$$

where : $f(x)$ is the d-dimensional feature of the test instance x provided by penultimate layer of the incremental model $\mathcal{M}_k$; $\mu_i = \frac{1}{n_i}\sum_{l=1}^{n_i} f(x_l)$ - the mean feature of the exemplars available for the i^{th} class. Note that, in order to reduce the majority bias, nem is performed after the selection of exemplars for new classes. nem calibration replaces the classification layer $\mathcal{C}_k$ of deep models by an external classifier which was explicitly designed to counter imbalance. Consequently, $iCaRL$ is not an end-to-end incremental learning method. The class predicted for test instance x after calibration is given by the $argmin$ function applied to the set of Euclidean distances computed for all classes using Eq. 6.4.

6.3.5 . Balanced fine tuning calibration (bal)

As an alternative to $iCaRL$ [188], the authors of [25] propose an end-to-end incremental learning method. The bias in favor of majority classes is reduced by introducing a second training step. After the initial training which creates $\mathcal{M}_k$ using the imbalanced dataset $\mathcal{D}_{N_k}$, a model $\mathcal{M}_k^{bal} : \mathcal{X}_{N_k}^{bal} \rightarrow \mathcal{Y}_{N_k}$ is trained. $\mathcal{M}_k^{bal}$ exploits $\mathcal{D}_{N_k}^{bal}$ a balanced version of $\mathcal{D}_{N_k}$ which includes $\frac{B}{N_k}$ exemplar images for both old and new classes and is fine tuned starting from $\mathcal{M}_k$. We modify the approach slightly in that balanced fine tuning only learns the weights of the classification layer $\mathcal{C}_k^{bal}$, instead of fine tuning the entire model. This modification is done in order to make bal calibration more comparable to the other calibration methods, which do not modify the feature extractor the deep model. It is also motivated by the fact that initial experiments run with full fine tuning of $\mathcal{M}_k$ provided lower results than fine tuning only the classification layer. Note that bal has a higher computational cost at training time since it requires a supplementary training step. The calibrated prediction of the i^{th} class obtained with bal can be written as :

$$\hat{y_i}^{bal} = \mathcal{C}_k^{bal}(i) \tag{6.5}$$

where $\mathcal{C}_k^{bal}$ is the output of classification layer of the balanced model $\mathcal{M}_k^{bal}$ for the i^{th} class. The class predicted after calibration is the $argmax$ value obtained over all classes using Eq. 6.5.

6.3.6 . Batch mean based calibration (mb)

The analysis of raw classification scores from Figure 6.1 provides support for a bias in favor of new classes in imbalanced incremental learning. A simple way to reduce this imbalance is to exploit the mean prediction scores of new and old classes of incremental state k. The calibrated score of the i^{th} class is written as :

$$\hat{y_i}^{mb} = \frac{\mu_{new}}{\mu_{old}} \hat{y_i} \tag{6.6}$$

where the means are defined as
$\mu_{new} = \frac{1}{\sum_{l=1}^{P_k} n_l} \sum_{l=1}^{P_k} \sum_{q=1}^{n_l} \hat{y_q}$ and
$\mu_{old} = \frac{1}{\sum_{l=1}^{N_{k-1}} n_l} \sum_{l=1}^{N_{k-1}} \sum_{q=1}^{n_l} \hat{y_q}$ for new and old classes respectively. Note that here we hold out validation sets for new and old classes in order to compute their mean classification scores. The class predicted after calibration is the $argmax$ prediction value obtained after applying Eq. 6.6.

6.3.7 . Fisher-Jenks based calibration

The mean based calibration operates at incremental batch level. It disregards the fact that, due to dataset imbalance, some of the new classes might fall in the minority classes set. To counter this problem, we propose a calibration method which makes use of class image counts and of their associated classification score. We use the Fisher-Jenks natural breaks method [110] to group classes. This method ensures an optimal distribution of a set of values in a predefined set of L clusters. It is thus appropriate to deal with the different imbalance profiles that occur in imbalanced incremental learning. In our case, the inputs given to Fisher-Jenks are the image counts n_i associated to the N_k classes learned in incremental state k. The calibrated score of the i^{th} class is written as :

$$\hat{y_i}^{fj} = \frac{\mu_{cl_L}}{\mu_{cl(i)}} \hat{y_i} \tag{6.7}$$

where $\mu_{cl(i)}$ is the mean prediction score of the Fisher-Jenks cluster which includes the i^{th} class and μ_{cl_L} is the mean prediction score of the L^{th} cluster with the largest number of instances per class. Similar to the mb method from Subsection 6.3.6, the means are computed using a validation set. The number of Fisher-Jenks clusters is set using a cross-validation with the validation set. The class predicted after calibration is the $argmax$ prediction value obtained by applying Eq. 6.7 to all initial class predictions.

6.4 . Evaluation

The experiments are designed to evaluate both dataset and incremental imbalances. All methods are evaluated with three large datasets designed for object, face and landmark recognition. $Soft$ and $strong$ imbalance configurations are created

to evaluate dataset imbalance. Three bounded memory size are introduced for each dataset in order to test the robustness of calibration method with respect to this central parameter of incremental algorithms.

6.4.1 . Baselines

The calibration methods studied here are applied on top of a vanilla fine tuning backbone which is run iteratively for each incremental state in order to integrate new classes. Naturally, vanilla fine tuning (FT hereafter) is the main baseline used here. The selection of exemplars is based on the herding mechanism [155]. To evaluate the usefulness of the proposed approach, we compare it to three competitive incremental learning methods :

— $iCaRL$ [188] combines classification and distillation losses to counter catastrophic forgetting and uses a nearest exemplar mean classifier to counter imbalance between past and new classes.

— BiC [238] introduces a linear layer at the end of the classification process to ensure fairness between past and new classes. A distillation term which is closer to the original formulation from [93] compared to $iCaRL$ is equally used.

— $LUCIR$ [97] proposes a combination of three elements to improve incremental learning. Cosine normalization is used for balancing the magnitudes of past and new class predictions. The distillation term is improved by handling feature vectors instead of raw scores. Finally, inter-class separation is favored in order to better separate embeddings of past and new classes.

6.4.2 . Datasets and methodology

We evaluate the baselines and the calibration methods on the following datasets :

— $ILSVRC$ [193] is a subset of 1000 $ImageNet$ classes used in the $ImageNetLSVRC$ challenges.

— $VGGFace2$ [24] ($VGGF2$ below) focuses on face recognition. We select the 1000 classes with the largest number of associated images.

— $GoogleLandmarks$ [167] ($LAND$ below) was built for landmark recognition and we again select 1000 classes with the largest number of associated images.

The test sets include 50000 images for $ILSVRC$ and $VGGF2$ and 20000 for $LAND$. There are 50 images per class for the first two datasets and 20 for the latter.

The original amount of imbalance in these three datasets is weak, as shown in Table 6.1. We introduce two imbalance configurations to evaluate behavior of the algorithms with different degrees of dataset imbalance :

— $soft$ - randomly retains between 50 and the initial number of images for each class.

— $strong$ - randomly retains between : 10 and 25 images for 300 classes, 26

	μ_{orig}	σ_{orig}	μ_{soft}	σ_{soft}	μ_{strong}	σ_{strong}
$ILSVRC$	1231	70	649	354	147	231
$VGGF2$	492	49	266	129	97	120
$LAND$	374	103	212	111	85	90

Table 6.1 – Means and standard deviations of image counts in the original datasets ($orig$) and the two imbalance configurations ($soft$ and $strong$).

and 75 for 300 classes, 76 and 100 for 200 classes and between 101 and the initial number of images for the remaining 200 classes.

The corresponding means and standard deviations are reported in Table 6.1. In the $soft$ configuration, slightly more than half of the original training data is kept and the standard deviation amounts to over 50% of dataset means. With $strong$, we discard a wide majority of original data and the resulting imbalance is much stronger and the standard deviation becomes higher than the mean in each case.

The evaluated calibration methods operate either at class level (iso, pl, th, nem, bal) or at an aggregate level which includes a subsets of the learned classes (mb, fj). For class level methods, we reuse the training images from the initial dataset as inputs for calibration. This is necessary since the number of available images is reduced, especially for old and/or minority classes and most of the methods require a rather large amount of data to provide reliable results. Consequently, the use of a validation subset would be suboptimal here. When inputs from different classes are aggregated in batches (mb and fj), the use of a proper validation split becomes possible. We create validation sets using 10% of the training data of old and new classes. We maintain the val/train split in the bounded memory B to avoid mixing the training and validation exemplars in different incremental states. Note that the outputs of C_k are used either in their raw form (iso, pl,mb, fj) or after transformation in probabilities by applying $softmax$ (th). This choice is made in order obtain an optimal configuration of each algorithm.

The experimental setup is inspired by the one proposed in $iCaRL$ [188]. Each dataset of 1000 classes is split into $k = 10$ incremental states. Each incremental state adds a batch of $P_k = 100$ classes to those that were already learned in states 1 to $k - 1$. The same class ordering provided in $iCaRL$ [188] is reused for $ILSVRC$ and a random ordering of classes is created to form $VGGF2$ and $LAND$ states. The size of bounded memory B was shown to have a central importance for the performance of incremental learning algorithms [25, 188]. To assess its influence on the proposed calibration methods, we report results with $B = \{5000, 10000, 20000\}$ exemplars stored in memory for each dataset and imbalance configuration.

A ResNet-18 architecture [88] is used as a backbone for all experiments. Re-

sNet have been successful in allowing neural nets to be deeper by tackling the problem to stagnation of performance with addition of layers after some point. They employ residual mapping as the basis function which adds the input values to approximate the final function. ResNet-18 has one (7*7) and sixteen (3*3) convolutional layers in addition to two max pooling layers and a final linear classification layer. We used the publicly available $iCaRL$ TensorFlow implementation in [188] with a binary cross-entropy loss and the original parameters proposed there. Vanilla fine tuning (FT) was implemented in Pytorch [176] using cross-entropy loss. The models were trained for 25 epochs with a initial learning rate of 0.1 at every incremental state and scheduled to decay by 0.1 when the loss plateaus out for 5 epochs. For $VGGF2$, face cropping is done with MTCNN [249] before further processing. Training images are processed using randomly resized 224×224 crops and horizontal flipping and are normalized afterwards.

6.4.3 . Metrics

— Accuracy - the performance of different methods is evaluated using top-1 accuracy for each incremental step defined as :

$$acc = 100 * \frac{1}{n} \sum_{i=1}^{n} argmax(\hat{Y}) == y_i \tag{6.8}$$

, where $\hat{Y}$ is the set of predicted score and y_i is the true label for test sample i. This measure is then averaged over all incremental states in order to obtain a single value for the entire incremental process. Note that averaged accuracy is the usual metric employed in incremental learning [97, 188, 238]. The test dataset contains the same number of samples for each class. This gives equal importance to all the classes irrespective of class-distribution in the training dataset. The test sets include 50000 images for $ILSVRC$ and $VGGF2$ and 20000 for $LAND$. There are 50 images per class for the first two datasets and 20 for the latter.

— Expected Calibration Error (ECE)- is a metric to ascertain the difference between the model accuracy and confidence [73]. The estimation of accuracy and confidence is done by dividing the samples into bins based on confidence. In our implementation the number of bins M are set to 20, to give 20 intervals of 1/M = 0.05 size from 0 to 1. B_m are set of samples in the interval m , with m = {1,2 ... M} and n is the number of samples in the test dataset.

$$ECE = \sum_{m=1}^{M} B_m/n * ||conf(B_m) - acc(B_m)|| \tag{6.9}$$

$$conf(B_m) = 1/B_m \sum_{i \in B_m} max(\hat{P}_i) \tag{6.10}$$

$$acc(B_m) = 1/B_m \sum_{i \in B_m} 1(argmax(\hat{Y}_i) == y_i) \tag{6.11}$$

,where $\hat{P}_i$ and $\hat{Y}_i$ are the set of predicted probability and score respectively and y_i is the true label for test sample i. The values of ECE range from 0 to 1, with lower values indicating better model calibration.

6.4.4 . Analysis of Accuracy of Calibration methods

B	Dataset	$iCaRL$	$LUCIR$	BIC	FT	FT_{is}	FT_{pl}	FT_{th}	FT_{nem}	FT_{bal}	FT_{mb}	FT_{fj}
5000	$ILSVRC$	21.8	**45.5**	41.3	38.7	23.5	31.4	45.0	39.3	40.5	41.6	41.5
	$VGGF2$	61.0	84.4	78.7	81.4	42.7	65.5	**85.4**	81.9	81.4	84.9	85.1
	$LAND$	64.1	86.9	80.6	84.3	37.9	76.0	**88.0**	85.2	81.1	85.7	86.0
10000	$ILSVRC$	23.6	48.9	45.5	45.3	32.1	38.6	**49.8**	44.8	45.6	46.9	46.4
	$VGGF2$	62.1	86.9	80.3	86	66.1	76.5	**88.0**	85.2	82.2	87.4	87.7
	$LAND$	65.7	88.9	82.1	88.9	53.9	84.6	**90.7**	88.2	85.8	89	89.2
20000	$ILSVRC$	24.5	52.7	49.7	50.1	38.3	44.8	**53.4**	48.6	49.8	50.7	50.3
	$VGGF2$	62.2	88.4	81.6	90.2	80	86.1	**91.0**	88.8	87.3	90.5	90.9
	$LAND$	65.8	90.8	83.3	92.2	75.4	90.5	**92.6**	90.8	91.6	92.0	92.1

Table 6.2 – Top-1 average accuracy for the $soft$ imbalance configuration and $B = \{5000, 10000, 20000\}$ bounded memory sizes. The first two columns represent $iCaRL$ [188] and vanilla fine tuning (FT), our baselines. The next two columns are calibrated versions of FT as follows : FT_{iso} - isotonic regression; FT_{pl} - Platt scaling; FT_{th} - thresholding; FT_{nem} - nearest-mean-of-exemplars; FT_{bal} - balanced fine tuning; FT_{mb} - batch mean based calibration; FT_{fj} - Fisher-Jenks based calibration. Following [25], accuracy scores are averaged over the incremental states of the system and the first, non-incremental, state is ignored.

The obtained results are presented in Tables 6.2 and 6.3. A detailed view of top-1 accuracy for the incremental states of $ILSVRC$ with $B = 5000$ bounded memory for $soft$ and $strong$ imbalance configurations is provided in Figure 6.2.

The performance level of the presented methods is much lower than that of non-incremental and balanced learning. We trained a ResNet-18 non- incrementally and using the full $ILSVRC$ dataset and obtained a top-1 accuracy of 73.0%. The non-calibrated accuracy (FT) obtained for $ILSVRC$ with memory $B = 20000$ are 50.1% and 37.5% for $soft$ and $strong$ imbalance configurations. The best results obtained for the same settings after calibration are 53.4% and 39.4% respectively. If the allowed memory is $B = 5000$, performance goes from 38.7% and 29.9% (non-

B	Dataset	$iCaRL$	$LUCIR$	BIC	FT	FT_{is}	FT_{pl}	FT_{th}	FT_{nem}	FT_{bal}	FT_{mb}	FT_{fj}
5000	$ILSVRC$	13.9	21.7	24.5	29.9	17.8	23.6	**33.1**	31.0	29.2	28.9	30.2
	$VGGF2$	50.4	66.3	65.9	76.7	32.3	63.6	**78.6**	75.1	72.0	77.4	78.1
	$LAND$	57.5	77.2	73.3	80.8	36.6	74.7	**82.4**	80.7	80.4	80.7	81.3
10000	$ILSVRC$	15.9	24.9	26.2	34.4	24.3	29.1	**36.8**	34.9	32.1	33.0	34.3
	$VGGF2$	51.3	68.5	67.9	80.8	55.9	72.9	**81.7**	78.7	77.0	80.5	81.43
	$LAND$	58.8	79.4	74.9	85.7	47	82.6	**86.3**	84.6	84.4	85.3	85.8
20000	$ILSVRC$	16.2	27.0	27.1	37.5	29.1	33.0	**39.4**	37.6	34.7	36.2	37.3
	$VGGF2$	51.4	71.8	68.8	83.9	68.6	78.6	**84.6**	82.1	81.1	83.4	84.3
	$LAND$	60.4	80.9	75.1	87.8	65.8	85.5	**88.4**	86.4	86.0	87.5	88.1

Table 6.3 – Top-1 average accuracy for the $strong$ imbalance configuration and $B = \{5000, 10000, 20000\}$ bounded memory sizes. See Table 6.2 for the description of the different methods presented.

calibrated FT) to 45.0% and 33.1% (FT_{th}) for $soft$ and $strong$ configurations respectively.

Intuitively, performance for $soft$ imbalance (Table 6.2) is higher compared to that for $strong$ imbalance (Table 6.3). The difference between the two configurations is largest for $ILSVRC$, the most difficult dataset among the three tested. With a memory of $B = 10000$ exemplars, the difference in performance between $soft$ and $strong$ configurations for FT is 10.9%, 5.2% and 3.2% for $ILSVRC$, $VGGF2$ and $LAND$ respectively. The size of the memory has also a strong influence on results. For instance, the performance of FT on $ILSVRC$ for bounded memories $B = \{5000, 10000, 20000\}$ reaches 38.7%, 45.3% and 50.1% in the $soft$ imbalance configuration.

The combined effect of incremental learning and dataset imbalance is thus strong and, while calibration is useful, the problem remains an open one. The difference between $soft$ and $strong$ imbalance configurations is also well illustrated in Figure 6.2. These detailed results show that the induced imbalance has a particularly important effect in early incremental states. This is normal since the importance of dataset imbalance is reduced in later incremental states, where the incremental imbalance due to the bounded memory B acts upon a large majority of classes.

The analysis of individual calibration methods shows that isotonic regression (FT_{iso}) and Platt calibration (FT_{pl}) have detrimental effect for both $soft$ and $strong$ imbalance configurations. Both methods rely heavily on the number of available class samples. The negative influence of iso and pl is larger for lower memory sizes and, within each B size, for later incremental states. This is probably an effect of lack of sufficient data in order to obtain a stable parametrization of

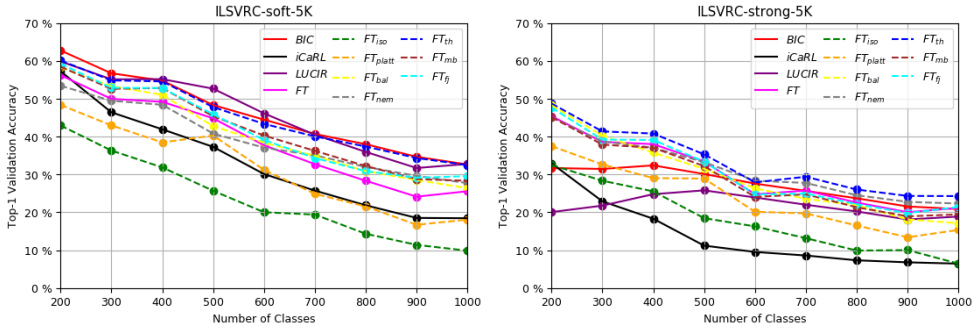


Figure 6.2 – Top-1 accuracy for $ILSVRC$ with memory $B = 5000$ in $soft$ (left) and $strong$ (right) imbalance configurations. To be aligned with the results from Tables 6.2 and 6.3, only the incremental states are represented. (*Best viewed in color.*)

the methods. The behavior of iso and pl in imbalanced incremental learning is different from the one previously reported in [165]. There are two main differences between the two studies : (1) the algorithms used are different (deep models here and shallow models in [165]) and (2) the amount of data available for calibration which is much smaller here.

Thresholding (FT_{th}) improves performance for all tested configurations. This method has the largest positive effect among all methods tested in a wide majority of cases. th performs score post-processing and is less dependent of the number of samples than iso and pl. It provides the largest improvements for $B = 5000$, the memory setting which corresponds to the largest imbalance for the three visual tasks with $soft$ and $strong$ configurations. The results obtained for th confirm those presented in [23] for imbalanced learning. They indicate that this simple calibration method should be considered in priority for the implementation of imbalanced incremental learning applications.

Nearest-mean-of-exemplars (FT_{nem}) has contrasted performance. The method is beneficial for the object recognition task ($ILSVRC$), although with lower effect for $B = 20000$. For face recognition task ($VGGF2$) and landmarks ($LAND$) gain is observed only for $soft$ imbalance at 5000 budget . For $ILSVRC$, nem is more useful for the $strong$ imbalance configuration than for the $soft$ one. The method works on top of the penultimate layer of the deep model but is highly dependent of the number of samples available to compute the individual class means, a property shared with iso and pl. Note also that FT_{nem} is equivalent to an $iCaRL$ version in which the distillation loss was ablated. The authors of $iCaRL$ [188] report that nem classification has positive influence over a direct use of deep model predictions in all configurations tested in their paper. A main difference is that those tests were done with classification and distillation losses, with larger memory and with datasets that are initially balanced. The effect of nem is more contrasted for the imbalanced datasets tested here with a vanilla fine

B	Dataset	FT	FT_{is}	FT_{pl}	FT_{th}	FT_{nem}	FT_{bal}	FT_{mb}	FT_{fj}
5000	$ILSVRC$	**0.214**	0.231	0.310	0.545	0.389	0.252	0.216	0.218
	$VGGF2$	0.045	0.422	0.651	0.138	0.814	0.102	**0.033**	0.036
	$LAND$	0.013	0.373	0.756	0.118	0.843	0.130	**0.009**	0.011
10000	$ILSVRC$	**0.201**	0.317	0.382	0.497	0.444	0.266	0.213	0.202
	$VGGF2$	0.022	0.803	0.857	0.087	0.883	0.089	**0.021**	0.022
	$LAND$	0.011	0.533	0.841	0.091	0.874	0.111	**0.009**	0.012
20000	$ILSVRC$	0.172	0.379	0.444	0.462	0.482	0.271	0.197	**0.171**
	$VGGF2$	0.031	0.656	0.761	0.115	0.847	0.107	**0.028**	0.029
	$LAND$	0.011	0.748	0.900	0.070	0.900	0.086	**0.007**	0.011

Table 6.4 – Expected Calibration Error for the $soft$ imbalance configuration and $B = \{5000, 10000, 20000\}$ bounded memory sizes. The first columns represent vanilla fine tuning (FT), our baseline. The next columns are calibrated versions of FT as follows : FT_{iso} - isotonic regression; FT_{pl} - Platt scaling; FT_{th} - thresholding; FT_{nem} - nearest-mean-of-exemplars; FT_{bal} - balanced fine tuning; FT_{mb} - batch mean based calibration; FT_{fj} - Fisher-Jenks based calibration. ECE are averaged over the incremental states of the system and the first non-incremental state is ignored.

tuning backbone with exemplars selected based on moving mean.

Balanced fine tuning (FT_{bal}) has a negative effect in most configurations. The methods provides improvement over FT for $ILSVRC$ at 5000 and 10000 budgets with $soft$ imbalance. The effect is particularly detrimental for $strong$ imbalance, Note that the reported bal performance is obtained by fine tuning only the classification layer of the incremental deep models $\mathcal{M}_k$. Balanced fine tuning (FT_{bal}) is performed by creating a balanced dataset, leading to much smaller datasets, especially if there is more dataset imbalance in addition to incremental imbalance. This would explain the sub-optimal performance as the dataset imbalance is increased.

Batch mean based calibration (FT_{mb}) improves performance over FT for all settings with $soft$ imbalance, while being comparative to FT for $strong$ imbalance. As for th , the gains are larger for lower memory size and for $ILSVRC$, the hardest visual task tested here. mb and fj have comparable results for $soft$ imbalance configurations, while fj gives slightly better results for $strong$ imbalance. mb is the simplest of all calibration methods tested since it only exploits mean predictions for old and new classes. It only accounts for the incremental imbalance as it groups new and old classes together, regardless of their image counts.

Fisher-Jenks based calibration (FT_{fj}) is a refined version of mb in which both the incremental and dataset imbalance are taken into account when clustering classes. The advantage of such clustering is more obvious for $strong$ imbalance configurations, where the dataset imbalance is more important compared to $soft$

B	Dataset	FT	FT_{is}	FT_{pl}	FT_{th}	FT_{nem}	FT_{bal}	FT_{mb}	FT_{fj}
5000	$ILSVRC$	0.276	0.239	0.288	0.627	0.345	0.426	0.339	**0.275**
	$VGGF2$	**0.060**	0.318	0.632	0.204	0.746	0.197	0.063	0.068
	$LAND$	0.018	0.360	0.743	0.165	0.799	0.179	**0.017**	0.018
10000	$ILSVRC$	**0.286**	0.174	0.233	0.662	0.307	0.405	0.337	0.288
	$VGGF2$	**0.044**	0.554	0.725	0.175	0.782	0.174	0.057	0.054
	$LAND$	0.015	0.465	0.822	0.129	0.838	0.157	**0.012**	0.018
20000	$ILSVRC$	0.263	0.287	0.326	0.601	0.372	0.453	0.324	**0.260**
	$VGGF2$	**0.044**	0.681	0.783	0.148	0.816	0.157	0.055	0.051
	$LAND$	0.014	0.653	0.850	0.110	0.856	0.127	**0.010**	0.015

Table 6.5 – Expected Calibration Error for the $strong$ imbalance configuration and $B = \{5000, 10000, 20000\}$ bounded memory sizes.

imbalance. Its performance is better than that of bal and nem for both $soft$ and $strong$ imbalance. ft globally has lower performance than th calibration.

A statistical analysis of the $LUCIR$ and FT_{th} reveal that FT_{th} is significantly better for $strong$ imbalance regime as compared to $soft$ imbalance. We compute the p-values over the accuracies at each incremental batch to ascertain the significance in the incremental setting. For $ILSVRC$ dataset, the p-value at budgets 5000, 10000 and 20000 are 0.91 , 0.84 and 0.85 for $soft$ imbalance as compared to 0.02 , 0.006 and 0.005 for $strong$ imbalance. Similarly for $Land$ dataset the p-value for $soft$ imbalance is at 0.43, 0.10 and 0.001 as compared to 0.001, 0.003 and 0.0028 for $strong$ regime. For $VGGF2$, the p-values for $soft$ imbalance is at 0.119, 0.038 and 0.019 as compared to 0.0041, 0.0042 and 0.0017 for $strong$ regime.

A final interesting observation is that $iCaRL$ performance lags well behind that of FT for all datasets and tested configurations. Further, FT baseline is competitive with $LUCIR$ and BIC, at $soft$ imbalance, while being clearly the preferable option in the $strong$ imbalance regime. This comparison is contrary to the conclusions of [188], where FT has significantly worse performance compared to $iCaRL$. However, that evaluation was biased insofar $iCaRL$ was using a memory of past classes while FT results obtained in absence of this memory. Our results indicate that, when running a fair comparison, the simpler FT method is clearly a better suited backbone for incremental learning with bounded memory than the state-of-the-art backbone which combines classification and distillation losses [25, 109, 188].

6.4.5 . Analysis of Expected Calibration Error

The results for Expected Calibration Error ECE for the calibration methods are presented in Tables 6.4 and 6.5. The first main observation is that the value of ECE for FT is higher for $ILSVRC$ dataset as compared to the other two

datasets. This is explained by the fact that $VGGF2$ and $LAND$ are easier to learn and FT provides significantly higher accuracy for these two datasets. Hence, the confidence is matched with high performance, which is not the case for $ILSVRC$.

Isotonic regression (FT_{iso}) and Platt calibration (FT_{pl}) provide very high ECE values as compared to FT, particularly for $VGGF2$ and $LAND$ datasets. A look at $LAND$ at $soft$ imbalance with 10000 budget, shows the accuracy at 53.9% and 84.6% for FT_{iso} and FT_{pl}, whereas the ECE is 0.533 and 0.841 respectively. This allows us to infer that the confidence of probabilities after FT_{iso} and FT_{pl} calibration is quite low, and the models actually under-calibrated. This can be partly explained by limited number of positives instances for a class, and high number of negative instances in one-vs-all calibration used in Isotonic Regression and Platt Scaling.

The results for FT_{nem} are similar to FT_{iso} and FT_{pl} with very high values of ECE, particularly when the accuracy is high. We draw similar conclusions that the model is under-calibrated for FT_{nem} as well. Note that for FT_{nem}, the scores are calculated as the inverse of the distance to the class mean in the feature space, which are then used to derive the probabilities using the softmax function. The accuracy for FT_{bal} are slightly lower than FT, and this is also reflected in ECE values of FT_{bal} which are slightly higher than FT.

FT_{th} provides the most improvement in accuracy out of all the calibration method. th mitigate the bias towards minority classes by calibrating the score for a class depending on the number of samples in the given class. It provides better performance by increasing the confidence of minority classes, though it also makes the model more mis-calibrated. ECE score for FT_{th} is consistently higher than its FT counterpart. This shows that FT_{th} provides improvement in overall accuracy, but at the some expense of calibration of model.

The proposed methods FT_{mb} and FT_{fj} provides the best calibration out of all the calibration methods. The calibration is quite similar to FT with ECE values being quite close to ones for for FT. This is an interesting results since FT_{mb} and FT_{fj} are the only methods which provide improvement in overall accuracy while not adversely affecting the calibration of the model.

6.5 . Conclusion

We performed a study of score calibration methods in an incremental and imbalanced deep learning setting which was not explored before. Calibration methods selected from both imbalanced and incremental learning streams of research were thoroughly compared using three visual tasks, two imbalance configurations and three bounded memory sizes for incremental learning. The obtained results indicate that, while calibration is certainly useful, imbalanced class incremental learning remains an open problem. They also show that both dataset imbalance and memory size have an important impact on performance. This is particularly true for object

recognition, the most difficult of the three tested tasks, and for lower memory sizes.

The performance of the evaluated calibration methods is variable. Isotonic regression and Platt calibration, which were shown to work well when enough data per class is available [165], have a negative effect on results here. This behavior is explained by the scarcity of available data when working in an incremental setting. Nearest-mean-of exemplars [188] and balanced fine tuning [25], the calibration methods introduced in recent incremental learning works, have contrasted and negative effects respectively. Note that, after initial experiments, an adaptation of balanced fine tuning was performed so as to fine tune only the classification layer instead of the entire network as done in [25]. The batch mean based and Fisher-Jenks calibration methods introduced here have a positive effect in most of the configurations. Fisher-Jenks behaves slightly better than mean based calibration. This is explained by the fact that the first method models both dataset and incremental imbalance while the second models only incremental imbalance. The best performance in terms of accuracy is obtained by thresholding based calibration, which uses the prior class probabilities to augment the scores of minority classes. An analysis of model calibration after the calibration method shows that overall FT_{mb} and FT_{fj} provide the best model calibration, while also tacking the imbalance.

Finally, the results also show that vanilla fine tuning is a better backbone for class incremental learning with bounded memory compared to a fine tuning which exploits both classification and distillation losses. The performance gap between the two approaches is significant and we advocate that future developments in class incremental learning should use vanilla fine tuning as baseline.

The reported results are interesting and we intend to develop our research along the following lines : (1) improve the vanilla fine tuning backbone using recent results in imbalanced learning [23] ; (2) explore other score calibration methods and (3) integrate incremental learning in content based multimedia retrieval frameworks.

7 - Conclusion and Perspectives

7.1 . Conclusion

In this work, we study two learning settings : active learning and incremental learning to tackle the data-dependent issues of deep neural networks. These two settings are important in the context of deploying deep learning models to real-world applications. Active learning is suitable to reduce the annotation cost of deep learning models. Incremental Learning is essential to create systems which evolve in dynamic domains. In both settings, we devise solutions to mitigate the effect of imbalance, while also tackling other open issues.

Large annotated datasets are a central requirement for training deep learning models in a supervised setting. The annotation of large datasets is both time and cost intensive. Active learning reduces this cost by the iterative selection of the most relevant samples based on model estimates on unlabelled data. A cold-start problem exists in AL which needs a large enough initial subset to be annotated to start the AL iterative process. In Chapter 3, we tackle the cold start problem by using an external dataset coming from a source domain. We propose a single stage setting, where the samples are selected using the knowledge from a source domain. This removes the need of a labeled dataset to kick-start the classical active learning setting, while also avoiding the time-consuming iterative training of the model. The objective is to make a diverse and balanced selection of samples, while discovering maximum of classes from a completely unlabeled dataset. The focus is on imbalanced dataset where random selection is clearly sub-optimal as it populates the selected set with samples from majority classes.

In Chapter 4, we extend our study on imbalanced datasets to the classical iterative active learning setting. The iterative setting becomes more pertinent when enough samples have been annotated for the target domain to provide reliable uncertainty estimates which are effective for AL task. In the iterative setting, we devise solutions to mitigate transfer of imbalance by applying balancing and diversity constraints. In particular, we propose a method which favors samples likely to be in minority classes so as to reduce the imbalance of the labeled subset and create a better representation for these classes. The evaluation is done with classical baselines and also with the method introduced in the single stage AL setting introduced in Chapter 3. In both the works, the proposed solutions are tested against state of the art techniques in active learning as well as in imbalanced learning.

In Chapter 5, we propose a new measure of informativeness based on the evolution of probability distribution between successive iterative states. The strategy prioritized samples which are predicted with high certainty in the previous iteration but which give uncertain prediction in the current model. This also allows to effectively assign a pseudo class to each sample depending on the confident predictions

of previous iteration. A diversification step was added to select samples from different regions of the classification space and thus introduces a representativeness component in our approach. Evaluation is done against competitive methods with three balanced and imbalanced datasets and outperforms them.

Here, we summarize the main conclusion from the works on active learning.

- **Single stage setting** We propose a single stage setting to tackle the cold start problem in active learning. The results show that single stage setting outperforms random sampling, which is normally used to select the initial subset in AL. The diversification process which selects samples with different source class prediction allows to select a diverse set of samples. Further we introduce a balancing step which is activated depending on the imbalance accumulated in the labeled set and the budget left. The balancing step focuses the labeling process on classes which are underrepresented in the annotated subset. Both adaptations have a positive effect as long as features are efficiently transferable between the source model and the target datasets. Further, we show that the proposed method helps to reduce the imbalance in the selected set, while also selecting samples from more number of classes. Finally, we test our methods on balanced datasets and show that the balancing step is beneficial for all acquisition function.
- **Imbalanced Datasets** Imbalance is big source of bias in dataset which leads to unfavourable predictions towards less represented classes. We test the affect of imbalance in both the single stage and iterative setting of AL. The results show that imbalance has to explicitly taken into account when creating the labeled dataset. Active learning techniques propagate the imbalance from unlabelled dataset while can also induce imbalance in the labeled set when working with balanced dataset. Further, this imbalance is only partially treated using techniques from imbalance learning such as thresholding [23]. This provides a strong motivation for designing techniques to mitigate the propagation of imbalance at the time of sample selection for annotation. AL could save considerable time and expertise in building datasets for real world applications. Real-world applications also normally contain imbalance with lesser occurrence of classes of interest. The balancing and diversification methods developed in our work help to create better datasets at lesser annotation cost.
- **Shallow classifiers over fixed representation** In our work with both single stage and iterative active learning setting, we test our methods at **low annotation budgets**. The training of deep models from scratch or the fine-tuning based approach requires large enough samples to be annotation to avoid overfitting and to provide efficient estimates for the AL task. We test the use of shallow classifiers over the representation of a pretrained model as an alternative to classical fine-tuning. Our results show that shallow classifier not only outperform fine-tuned model, but also the

probability estimates from shallow classifier are more reliable for AL task. This has important, insofar that almost all AL works explore only the more computationally expensive fine-tuning scheme, which require deep learning hardware. Alternatively, shallow classifiers over fixed representation can be easily trained on general purpose CPUs. Further, we also propose a cross-validation step to switch from shallow classifier to fine-tuning method when enough samples points are selected in Chapter 4.

- **Using iterative model probability** AL is usually implemented in an iterative cycle with model trained with more annotated data in each cycle. Thus it is possible to store the iterative model's estimates on unlabelled dataset with minimal cost. In Chapter 5, we propose a new measure of informativeness based on evaluating the change with every update of model. The proposed measures are more effective for shallow classifiers than the fine-tuning scheme. The fixed representation used for shallow classifiers allows the proposed measure to effectively evaluate the distance of the samples to the classifier boundary in the preceding and the current iterations. Further, we apply a diversification procedure to combine the informative and representative objectives of active learning. The efficiency of probability estimate from previous model is subject to the batch sizes used in AL. At large batch sizes, the estimates from the previous model might become less relevant.

In the work of incremental learning, Chapter 6, we tackle the problem of learning from a dynamic domain where classes are learned incrementally. We allow a fixed memory budget to store the examples from old classes. The model is fine-tuned whenever new classes are added using the exemplars from old classes and the data from new classes. We consider a real-world setting where the dataset can contain imbalance. This leads to two kind of imbalance in the dataset :

(1) imbalance between old and new classes due to fixed-memory incremental setting and

(2) imbalance inherent in the dataset.

We show that IL with fixed memory for exemplars from old classes can be solved as a imbalance learning problem. A wide range of post-processing calibration methods are considered for treating the two kind of imbalance. The best performance is generally obtained by thresholding based calibration, which uses the prior class probabilities to augment the scores of minority classes. An analysis of model calibration after the calibration method shows that proposed methods FT_{mb} and FT_{fj} provide the best model calibration, while also tacking the imbalance. The results also further show that vanilla fine tuning is a better backbone for class incremental learning with bounded memory compared to a fine tuning which exploits both classification and distillation losses.

7.2 . Future Works

The methods developed in our work can be tested and improved upon in the following ways :

- **AL setting selection** In our work, we test and compare both the single stage and iterative setting of active learning. Instead of randomly selecting the first batch of images as done in the iterative setting, single stage AL setting is used to select the initial dataset. The efficiency of the single stage setting depends on the similarity between source and target domain. Further, with enough number of samples the target model becomes more suitable for the active learning task. A method to select the best strategy can be explored which switches from single stage AL setting to the iterative setting. The switch point would depend on the budget and the re-usability of source domain for the target AL task. This would also include the cold-start problem to try to ascertain the number of samples required before active learning can be started. [58] provide some theoretical framework in this context to select the number of samples randomly before active learning can be performed.
- **Universality/ Domain Adaptation** The methods developed in single stage setting : diversification and balancing depends on transferability of features between source and target domain to create a mapping between source and target classes. These methods could benefit from a more universal representation [213] to be applicable to more diverse dataset. We also show that using shallow classifier over fixed representation is a better alternate to classical fine tuning method at low budgets. Universal representation could also help the shallow classifier scheme to be more effective, and outperform fine-tuning till higher budgets. Further, curating the features from the source domain to suit the target domain can be explored [221].
- **Using iterative models for AL** In Chapter 5, we propose a method to select samples which move from certain to uncertain regions with the update of iterative model. This idea can be extended to fine-tuning scheme by using estimates from different snapshots during the fine-tuning process. Such an approach, would help identify samples which the model finds difficult to learn and forgets easily. Further, previous works [28, 61] have shown that for easier samples intermediate models during the optimisation can provide better confidence estimates as compared to final model.
- **Calibration methods for incremental learning** We test a range of calibration methods to reduce the bias towards the minority classes and improve the model calibration. Application of these post-processing techniques on top of latest works in incremental learning would help to ascertain their broader applicability. To do this the vanilla fine tuning backbone can be improved using recent results [238, 97].

8 - Appendix

8.1 . Résumé en français

8.1.1 . Contexte : la reconnaissance visuelle à l'ère de l'apprentissage profond

Les algorithmes d'apprentissage profond et en particulier les modèles neuronaux profonds supervisés ont permis des progrès impressionnants au cours de la dernière décennie pour diverses tâches de reconnaissance visuelle telles que la classification, la détection d'objets ou la segmentation sémantique (voir la Figure 8.1 pour un aperçu rapide de ces tâches). En effet, pour toutes ces tâches, les performances, évaluées sur des benchmarks publics, ont franchi un cap grâce aux modèles neuronaux profonds. Par exemple, pour le défi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [194], la précision moyenne du top1 est passée de 50,9% en utilisant des modèles de pré-apprentissage profond en 2011 à 90,2% dans les travaux récents [53]. Un gain similaire est observé pour la tâche de détection d'objets sur MS-COCO [142], où la précision moyenne des boîtes englobantes est passée de 34,9% en 2015 avec des modèles R-CNN plus rapides [66] à 58,7% dans la méthode récente [148]. En particulier, l'apprentissage profond a fait évoluer le paradigme de l'utilisation de caractéristiques créées à la main vers l'apprentissage par représentation avec des modèles multicouches [16]. De plus, les réseaux de neurones profonds (DNN) se sont révélés efficaces pour apprendre de puissantes représentations hiérarchiques des données qui peuvent même être **transférées** à d'autres tâches [170].

Le **avènement des unités de traitement graphique (GPU)** [34, 232] pour correspondre à la nature intensive en calcul des algorithmes d'apprentissage profond, la meilleure conception des **architectures d'apprentissage profond** [125, 39, 206, 66, 89] et la disponibilité des **grands ensembles de données annotées** [193, 166, 142] sont quelques-uns des principaux facteurs qui expliquent les gains de performance et l'omniprésence de l'apprentissage profond. Par exemple, les réseaux de neurones convolutifs (CNN), qui constituent l'architecture de base de la plupart des tâches de vision par ordinateur aujourd'hui, ont été envisagés dès 1988 pour la tâche de classification des phénomes [6]. En 1989, Yann LeCun a utilisé les CNN pour la reconnaissance de caractères manuscrits et a formé un réseau neuronal à l'aide de l'algorithme de backpropagation [131]. L'utilisation des réseaux neuronaux convolutifs était limitée à l'époque par le matériel informatique disponible, ce qui a entraîné une interruption entre les travaux de LeCun et la résurgence des CNN au cours de la dernière décennie grâce à l'utilisation de GPU spécialisés. Les GPU, initialement développés pour les consoles de jeu, effectuent efficacement les calculs répétitifs nécessaires aux réseaux neuronaux et ont ainsi contribué à résoudre le goulot d'étranglement matériel des DNN [172] et ouvert

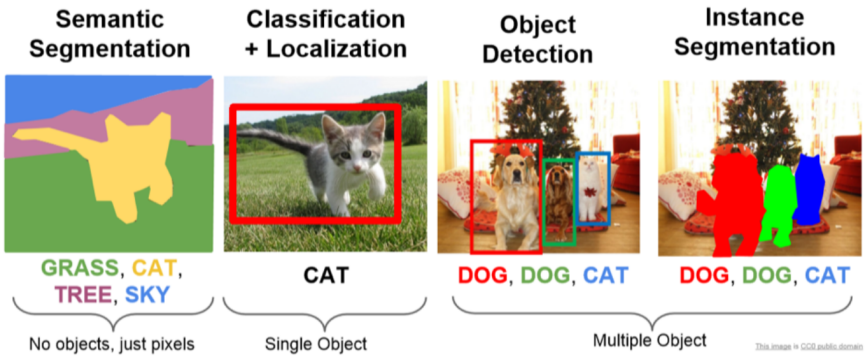


Figure 8.1 – Illustration de diverses tâches de vision par ordinateur pour lesquelles les réseaux de neurones profonds fournissent des performances de pointe [136].

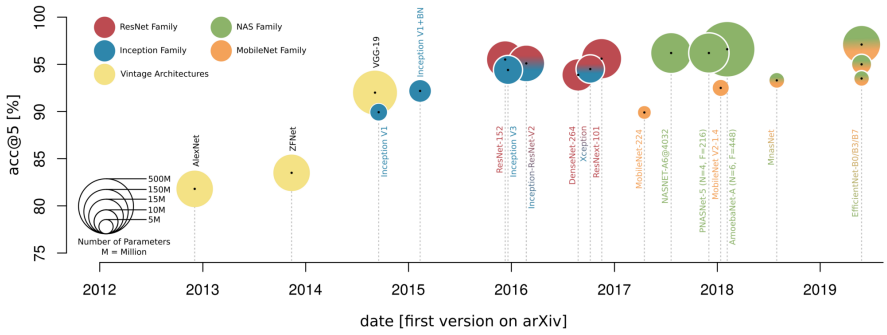


Figure 8.2 – Évolution de la taille et de la précision des modèles sur ILSVRC [95]. La taille des modèles a augmenté avec le temps, de même que leur précision.

l'ère des réseaux neuronaux profonds.

En particulier, les GPU ont permis des avancées dans la conception de l'architecture avec davantage de paramètres entraînables, ce qui a permis d'augmenter la capacité de représentation des réseaux neuronaux profonds. Comme le montre la figure 8.2, il existe une forte relation entre la taille des modèles profonds (nombre de paramètres) et leur efficacité en termes de précision de la tâche. Aujourd'hui, les architectures d'apprentissage profond comprennent des millions de paramètres qui doivent être optimisés pour une tâche donnée. Ce caractère surparamétré des modèles profonds est un atout pour l'apprentissage de représentations complexes, mais c'est aussi une des limites de ces modèles, puisqu'il limite leur interprétabilité [143]. En effet, si certains progrès ont été réalisés pour augmenter l'interprétabilité des modèles d'apprentissage profond [46], un compromis entre interprétabilité et per-

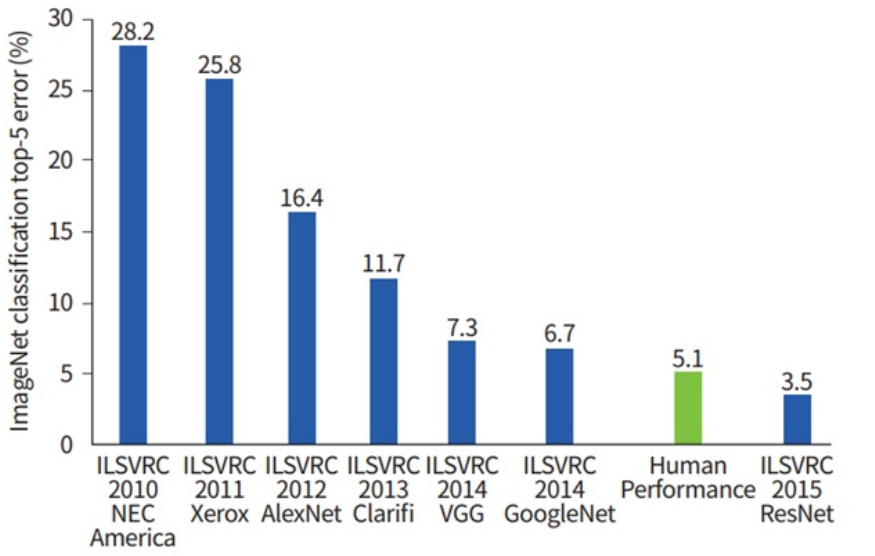


Figure 8.3 – Top-5 Taux d'erreur des lauréats annuels sur la tâche de classification ILSVRC [145]. L'architecture ResNet a surpassé la performance humaine de 5,1% en 2015.

formance a été établi en raison de la nature surparamétrée des grands modèles d'apprentissage profond [72].

Un autre facteur important qui explique la hausse de l'utilisation des modèles d'apprentissage profond est l'augmentation du nombre de jeux de données et de repères disponibles. À des fins de recherche, la communauté de la vision par ordinateur a développé de nombreux jeux de données annotés de grande taille et de haute qualité, tels que ImageNet [193] et CIFAR [124] pour la tâche de reconnaissance d'objets, MS COCO [142] et Open Images [128] pour la détection et la segmentation d'objets ainsi que la classification multi-label ou les Google landmarks [166] et la reconnaissance de scènes intérieures [253] pour la tâche de compréhension de scènes, entre autres. Les ensembles de données annotées sont construits à l'aide de ressources lexicales solides et riches, fournies par une supervision humaine au cours du processus d'annotation. Par conséquent, la construction de ces ensembles de données est coûteuse en temps et en ressources humaines. Par exemple, la base de données ImageNet est construite sur la ressource lexicale Wordnet bien connue [157] qui encode les connaissances de sens commun. Chacune des 14 millions d'images de la base de données ImageNet est affectée à l'une des 22 000 classes par un processus impliquant une annotation manuelle coûteuse [193].

Ces ensembles de données ont servi à la communauté pour évaluer et développer différentes approches. Par exemple, la longue histoire entre les modèles

profonds et le jeu de données ILSVRC est illustrée dans la figure 8.3 [145]. Bien que ces ensembles de données académiques aient permis à l'apprentissage profond de progresser sur plusieurs tâches difficiles, la disponibilité de grands ensembles de données parfaitement annotés dans la plupart des scénarios pratiques ne peut être supposée. Il s'agit d'un obstacle majeur pour les applications du monde réel, car la performance des modèles d'apprentissage profond est fortement liée à la taille de l'ensemble de données d'entraînement. Les auteurs de [211] montrent que pour les tâches de vision par ordinateur, les performances augmentent de façon logarithmique en fonction de la taille de l'ensemble de données. De plus, comme la taille des modèles profonds a augmenté pour améliorer leur capacité de représentation, des ensembles de données plus grands sont nécessaires pour éviter l'over-fitting où le modèle mémorise les données d'entraînement au lieu d'apprendre des modèles utiles à partir des données [149]. Ainsi, alors que l'augmentation de la taille des modèles et des ensembles de données a permis aux modèles d'apprentissage profond de surpasser largement les modèles d'apprentissage automatique traditionnels, la conception de solutions d'apprentissage profond est devenue fortement dépendante de la disponibilité de grands ensembles de données, ce qui génère certaines limitations importantes.

8.1.2 . Modèles neuronaux profonds dépendants des données : quelques limites

Malgré son succès, l'apprentissage profond présente plusieurs inconvénients potentiels et importants découlant de sa nature dépendante des données, en particulier dans le contexte de la conception et du déploiement de modèles profonds pour des applications réelles. Nous détaillons ces facteurs dans ce qui suit.

- **L'annotation des données est une tâche très coûteuse** .

  Alors que les collections d'images à grande échelle sont désormais largement disponibles, par exemple sur le Web-corpus, leur étiquetage manuel reste une tâche fastidieuse et coûteuse. Dans le contexte des données publiques et générales, une solution classique pour limiter le coût et le temps d'annotation est d'utiliser le crowd-sourcing, mais cela n'est pas possible dans les domaines qui nécessitent la disponibilité d'experts du domaine pour faire l'annotation comme l'imagerie médicale [186]. De plus, certaines tâches visuelles sont très exigeantes car elles nécessitent une annotation très précise comme par exemple pour la tâche de segmentation sémantique. Enfin, la qualité de l'annotation dépend aussi fortement de la capacité à collecter et préparer un jeu de données, représentatif de la tâche visée. Le coût de l'annotation est donc un problème important et une limitation majeure pour l'apprentissage supervisé, qui nécessite d'entraîner le modèle en utilisant des données d'entraînement entièrement annotées.

  Pour répondre à cette forte limitation des modèles supervisés, une solution naturelle est de développer des schémas d'apprentissage efficaces en termes

d'étiquettes qui évitent le besoin de grands ensembles de données annotées. En l'absence d'annotations, **l'apprentissage non supervisé** [146, 222] est utilisé pour former un modèle qui apprend la structure implicite de la distribution des données. Dans l'apprentissage **semi-supervisé**, une combinaison d'apprentissage supervisé et non supervisé est utilisée. L'idée principale est d'utiliser une petite quantité de données étiquetées (ou annotées) et d'exploiter une grande quantité de données non étiquetées. **L'apprentissage faiblement supervisé** [254] allège le problème de l'obtention d'ensembles de données étiquetées de haute qualité, coûteux ou peu pratiques, en supposant des annotations de faible qualité, c'est-à-dire des annotations inexactes, bruyantes ou incomplètes. À la frontière entre l'apprentissage semi-supervisé (petit ensemble de données annotées) et l'apprentissage faiblement supervisé (annotation incomplète), l'apprentissage actif propose de sélectionner une petite quantité de données pertinentes qui doivent être étiquetées afin d'avoir le plus grand impact possible sur l'apprentissage d'un modèle supervisé. La transférabilité des représentations profondes apprises, mentionnée ci-dessus, a également stimulé le développement des approches dites de **apprentissage par transfert** [236] et de **adaptation au domaine** [228]. Dans ces paradigmes, nous supposons la disponibilité d'un domaine source, pour lequel nous disposons d'un large ensemble de données annotées de haute qualité pour entraîner un modèle, qui est ensuite adapté à un domaine cible moins riche en termes de données annotées. Enfin, un autre cadre d'apprentissage récent et important pour éviter le besoin de grands ensembles de données annotées est l'apprentissage auto-supervisé. Il s'agit d'une rencontre entre l'apprentissage non supervisé et supervisé avec la construction automatique d'étiquettes pour les données non supervisées à l'aide de certaines tâches prétextes.

Ces différents schémas d'apprentissage répondent au coût d'annotation avec un succès variable. Le coût d'annotation des algorithmes non supervisés est faible, mais leur efficacité est limitée par l'hypothèse des clusters, qui suppose que les échantillons affectés à différents clusters sont sémantiquement différents [42], ce qui n'est souvent pas satisfait dans les applications du monde réel. Elle ne parvient pas non plus à capturer la sémantique des ensembles de données avec le même degré de raffinement et de performance que leurs homologues supervisés ou semi-supervisés [14]. Le coût de l'annotation reste important pour l'apprentissage semi-supervisé, les performances dépendant de la taille de l'ensemble de données étiquetées [76]. De plus, l'efficacité de l'approche semi-supervisée dépend également d'une hypothèse forte sur la sémantique des données (c'est-à-dire l'hypothèse du cluster, du collecteur et de la régularité) qui n'est pas nécessairement vérifiée sur des données réelles [223]. L'apprentissage par transfert et l'adaptation au domaine se sont avérés efficaces pour réduire le coût d'annotation,

mais ils sont limités par une *similarité* supposée entre le domaine source et le domaine cible [260]. Enfin, la performance de l'apprentissage auto-supervisé dépend de la capacité à concevoir une tâche prétextuelle efficace [111], ce qui peut être difficile pour les domaines de haute expertise.

Ainsi, nous soutenons que, malgré les progrès récents dans l'exploitation des données non étiquetées, il est crucial d'utiliser des données étiquetées de haute qualité dans l'apprentissage supervisé ou semi-supervisé. En effet, la plupart des avancées dans les tâches de vision par ordinateur utilisent une certaine forme de supervision parenciteforet2020sharpness,liu2021swin. En outre, l'expertise et les connaissances des experts du domaine peuvent être apportées au système d'apprentissage automatique au cours du processus d'annotation parencitehassanzadeh2011machine. L'annotation est également beaucoup plus importante dans les tâches de vision par ordinateur que dans les tâches de traitement du langage naturel, en raison du fossé sémantique bien connu [208].

- **L'annotation doit être un processus continu et dynamique** La plupart des ensembles de données peuvent être considérés comme statiques une fois constitués et les modèles d'apprentissage profond sont généralement appris sur des lots stationnaires de données d'entraînement. Néanmoins, dans les applications pratiques, de nouvelles données peuvent être acquises en permanence et les modèles d'apprentissage profond doivent donc tenir compte des situations dans lesquelles les informations deviennent disponibles de manière progressive au fil du temps. C'est le cas d'un grand nombre d'applications du monde réel. Par exemple, dans l'analyse de données [63] ou la robotique [51], le modèle doit s'adapter à l'environnement changeant. Dans les cas où l'on suppose que l'on a accès à toutes les données acquises précédemment, le problème devient trivial, bien que gourmand en temps et en ressources, car toutes les données acquises peuvent être utilisées pour entraîner le modèle en une seule tâche. Cette méthodologie est très inefficace et entrave également l'apprentissage de nouvelles données en temps réel. Lorsque l'accès aux anciennes données est limité ou impossible, le problème devient beaucoup plus compliqué. En particulier, les modèles d'apprentissage profond souffrent d'un **oubli catastrophique** [154] où les anciennes informations sont perdues lors du réentraînement pour apprendre de nouvelles informations. Si rien n'est fait pour empêcher ce phénomène, les prédictions pour les classes du passé deviennent aléatoires ou presque. C'est particulièrement vrai pour les algorithmes d'apprentissage profond qui dépendent fortement des données étiquetées. Il fait donc appel à des schémas d'annotation ou de formation qui tiennent compte de la nature dynamique du domaine visé.

  Dans la littérature, diverses approches ont été proposées pour aborder le problème des domaines dynamiques avec des motivations différentes. **Life-**

long learning ou apprentissage continu [174] apprend continuellement sur de nouvelles données tout en conservant les connaissances acquises dans le passé. **Les méthodes de méta apprentissage** [134] traitent une séquence de tâches, mais avec l'objectif de former un apprenant efficace sur une nouvelle tâche. Ainsi, les méthodes de méta-apprentissage tentent d'extraire des informations pendant la formation des tâches précédentes qui faciliteraient l'apprentissage de la nouvelle tâche. Cette stratégie d'apprentissage a été largement utilisée dans le contexte du **few shot learning** dans lequel nous voulons apprendre avec très peu d'échantillons [231]. Notez que les approches de méta-apprentissage sont différentes de celles de l'apprentissage tout au long de la vie, car elles ne mettent pas l'accent sur la rétention de la tâche précédemment apprise comme le fait ce dernier [30].

Les êtres humains et les animaux ont la capacité d'acquérir et d'étendre continuellement leurs connaissances en interagissant avec un environnement en constante évolution [22]. Cette capacité est essentielle pour concevoir des modèles qui s'améliorent avec le temps sans avoir à apprendre le modèle à partir de zéro à chaque fois que de nouvelles informations sont présentées [81]. C'est un domaine de recherche ouvert et une étape importante vers la création d'une intelligence générale artificielle [68].

- **Bias de l'ensemble des données** Une autre limitation importante due à la dépendance des modèles profonds vis-à-vis des données est leur grande sensibilité au biais dans les données. Plusieurs imperfections, telles que des étiquettes bruyantes ou des distributions déséquilibrées, peuvent constituer un biais dans les données. Le biais de jeu de données a récemment été mis en évidence dans les tâches de vision, principalement en raison des applications de reconnaissance des visages qui montrent un biais négatif de l'algorithme vers les catégories de la population qui sont moins représentées dans les jeux de données d'entraînement. Les auteurs in [116] fournissent une revue systématique de la littérature sur le problème du biais dans les logiciels de reconnaissance faciale et soulignent le rôle des données d'entraînement dans l'instillation du biais dans l'algorithme.

  Ce biais est également présent dans des domaines très contrôlés. Par exemple, dans le domaine médical, il a été démontré dans [129] que le biais de genre a un effet très fort dans le diagnostic médical assisté par ordinateur et qu'il est le facteur principal des prédictions sous-optimales pour le genre sous-représenté.

  Le déséquilibre des classes est un problème majeur qui est généralement négligé lorsque l'on travaille avec des ensembles de données universitaires. Ces ensembles de données peuvent être considérés comme optimisés pour l'apprentissage puisque leurs classes sont représentées de manière équilibrée, c'est-à-dire que le nombre d'instances de chaque classe dans l'ensemble de données d'apprentissage est équilibré. En pratique, il est pru-

dent de considérer que les ensembles de données sont toujours imparfaits. Ces imperfections peuvent résulter de problèmes dans le processus d'acquisition des données ou de diverses complexités inhérentes aux données de mots réels. Le déséquilibre des classes [85] apparaît lorsque certaines classes de l'ensemble de données sont surreprésentées ou sous-représentées par rapport aux autres classes. Les ensembles de données construits pour des applications réelles sont souvent déséquilibrés et les classes d'intérêt sont particulièrement sous-représentées par rapport aux autres classes qui sont fréquentes. Par exemple, dans le domaine de l'imagerie médicale, on rencontre un déséquilibre entre les cas pathologiques et les cas normaux, car les cas présentant des anomalies pathologiques peuvent être rares ou uniques [151]. L'apprentissage à partir de données déséquilibrées, c'est-à-dire avec des classes minoritaires et majoritaires, conduit à un biais de prédiction en faveur des classes majoritaires. Cet effet négatif est bien étudié pour les méthodes classiques d'apprentissage automatique, comme décrit dans les deux études suivantes [86, 108]. Une étude similaire [23] a été menée récemment sur les algorithmes d'apprentissage profond avec une conclusion similaire sur l'effet négatif du déséquilibre sur les performances de prédiction.

8.1.3 . Le déploiement de l'apprentissage automatique nécessite des schémas d'apprentissage itératifs

Dans les applications du monde réel qui déploient des modèles d'apprentissage automatique, il est courant de supposer un schéma itératif où les performances du modèle sont surveillées en permanence pendant le déploiement et peuvent être mises à jour avec les nouvelles données acquises, comme illustré dans la figure 8.4. Alors que cet aspect est souvent considéré dans le domaine appelé ML ops (Machine Learning Operational) dans lequel un cycle de vie ML est considéré, ce schéma itératif est rarement pris en compte dans la recherche où l'on ne considère que les trois étapes classiques de formation, validation et test. Dans cette thèse, nous étudions ce schéma itératif dans lequel nous supposons que les nouvelles données doivent être prises en compte de manière itérative afin de maintenir la performance du modèle.

Deux scénarios sont possibles lors de la mise à jour du modèle avec de nouvelles données :

1. les nouvelles données acquises ou annotées proviennent du même domaine (c'est-à-dire des mêmes classes sémantiques)
2. les nouvelles données proviennent d'un domaine différent où de nouvelles classes sont introduites.

Dans les deux cas, la prise en compte de nouvelles données est un défi dans le contexte des modèles profonds, en partie à cause des facteurs limitatifs dépendants des données décrits dans la section précédente. Nous proposons donc d'aborder ces

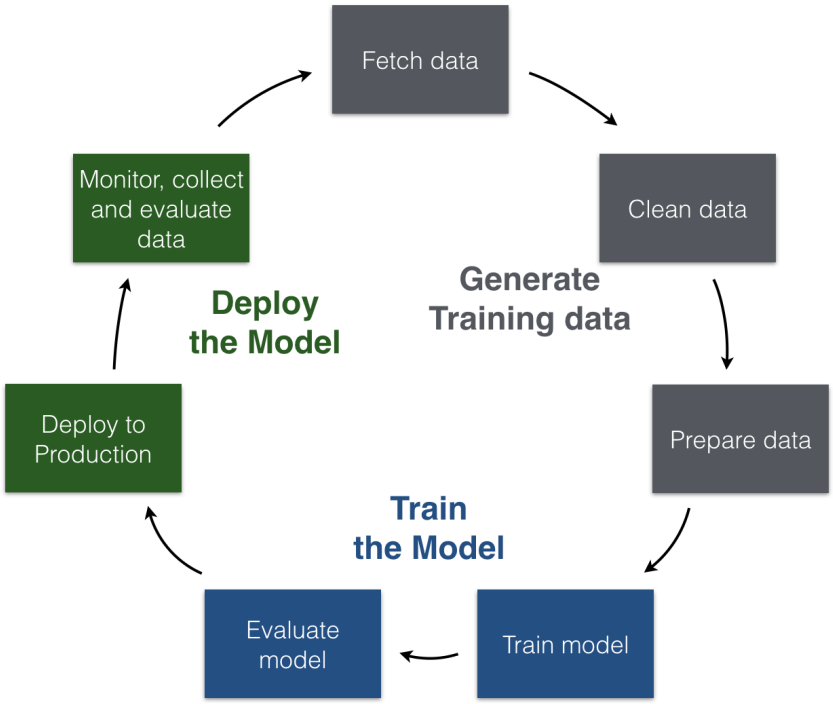


Figure 8.4 – Vue d'ensemble du pipeline d'apprentissage automatique. Il montre le schéma itératif couramment utilisé dans les applications pratiques. Les performances du modèle sont contrôlées pendant le déploiement et de nouvelles données supplémentaires peuvent être utilisées pour mettre à jour le modèle.

deux scénarios en adaptant deux schémas d'apprentissage itératif au contexte de l'apprentissage profond avec des données incrémentales, limitées et déséquilibrées. Pour aborder le premier scénario dans lequel des données nouvelles mais limitées d'un domaine donné doivent être annotées en permanence, nous nous basons sur le schéma **active learning**. Pour le second scénario, dans lequel les nouvelles données contiennent des échantillons de classes précédemment inconnues, nous nous basons sur le schéma **apprentissage incrémentiel de classe**. Nous présentons brièvement ces deux scénarios d'apprentissage itératif dans ce qui suit.

Apprentissage actif

L'apprentissage actif (AL) [199] tente de résoudre le problème du coût d'annotation de grands ensembles de données pour l'apprentissage supervisé. En partant du principe que tous les échantillons n'ont pas la même valeur pour le modèle, l'apprentissage actif tente de sélectionner les échantillons les plus importants pour une annotation manuelle. AL est généralement déployé de manière itérative. Un

nombre fixe d'échantillons est sélectionné par itération et annoté afin de réentraîner le modèle qui devient progressivement plus fort. La stratégie de sélection peut être conduite avec des objectifs différents mais complémentaires : maximiser la **informativité** où les échantillons qui sont susceptibles d'apporter de nouvelles informations sont sélectionnés [202, 32, 196, 15] ou la maximisation de la **représentativité** où le critère principal est d'assurer un ensemble diversifié d'échantillons afin d'apprendre une représentation forte de l'ensemble de données non étiquetées [198, 139, 36].

Récemment, AL a regagné de l'intérêt dans le contexte des modèles profonds. Par exemple, [189] fournit une étude récente sur l'apprentissage actif profond (DAL). Dans cette thèse, plusieurs aspects ont motivé l'utilisation du schéma AL. Tout d'abord, il répond à la fois au coût d'annotation et est bien adapté au schéma d'apprentissage itératif déployé dans les applications du monde réel. De plus, AL est un paradigme humain dans la boucle et permet d'ajouter continuellement une expertise de haut niveau dans le processus d'apprentissage par l'annotation. Il apporte également plus d'explicabilité [64] puisqu'il permet à l'expert d'observer l'évolution du modèle d'apprentissage et sa prédiction sur des données non étiquetées. Ainsi, avec l'aide de l'expertise humaine, il pourrait être possible de déterminer quels concepts ont été appris par le modèle et ce qui reste à apprendre. Enfin, dans l'AL, l'expertise est utilisée pour annoter les données brutes, contrairement à l'apprentissage non supervisé ou auto-supervisé où l'annotation est effectuée sur des données groupées, ce qui pourrait être biaisé [224]. Néanmoins, malgré ces travaux récents sur l'apprentissage actif profond, il reste des questions ouvertes qui limitent son utilisation dans des scénarios pratiques.

Apprentissage incrémentiel par classe

L'apprentissage incrémentiel par classe (CIL) [26] vise à ajouter de nouvelles classes au modèle d'apprentissage, tout en conservant l'efficacité pour les classes du passé. Comme nous l'avons vu précédemment, le principal défi de l'apprentissage dans des domaines dynamiques est l'oubli catastrophique [154] où les modèles perdent les connaissances acquises précédemment, lorsqu'ils sont réentraînés avec de nouvelles données. Il est donc nécessaire de réentraîner les modèles avec les données précédentes et nouvelles, ce qui entraîne un coût plus élevé en termes de ressources informatiques pour l'entraînement sur des ensembles de données plus importants, ainsi qu'en termes de mémoire pour le stockage de toutes les données antérieures.

Nous avons choisi de nous appuyer sur le CIL en raison de sa capacité à traiter les domaines dynamiques que l'on peut rencontrer dans les applications du monde réel. De plus, le CIL fait partie de l'objectif plus large de l'apprentissage continu ou tout au long de la vie où de nouvelles informations peuvent être continuellement assimilées dans le modèle. La réutilisation des informations apprises est donc essentielle pour limiter le coût de calcul de l'apprentissage du nouveau modèle à

partir de zéro avec les anciennes et les nouvelles données. En outre, le CIL permet également de réduire l'utilisation de la mémoire en empêchant ou en limitant la quantité d'instances passées à stocker. L'accès aux anciennes données peut être restreint ou impossible en raison de plusieurs facteurs tels que : la suppression des données sur le Web et dans le traitement des données en continu [77], la confidentialité dans le domaine médical [225] ou les ressources limitées dans les systèmes embarqués [175]. Ainsi, l'apprentissage incrémentiel est hautement souhaitable dans ces domaines dynamiques afin de stimuler la réutilisation des connaissances apprises pour une utilisation efficace des ressources, tout en réduisant également la dépendance aux instances passées.

8.1.4 . Contributions

Dans cette thèse, notre objectif est de fournir de nouveaux outils méthodologiques pour répondre à plusieurs limitations qui affectent le déploiement de modèles neuronaux profonds dans des applications de mots réels. Ces limitations sont au nombre de trois : le besoin de grandes données annotées pour construire des modèles efficaces et conscients du domaine, le besoin de schémas d'apprentissage itératifs afin de prendre en compte le comportement dynamique d'une majorité d'applications de mots réels et la nature déséquilibrée de la plupart des jeux de données du monde réel. Sur la base des deux schémas d'apprentissage itératif présentés dans la section précédente, le schéma d'apprentissage actif et celui d'apprentissage incrémentiel par classe, nous appliquons et évaluons nos solutions sur des tâches de reconnaissance visuelle telles que la classification d'images, mais nos solutions sont génériques et pourraient être appliquées avec une petite adaptation à d'autres tâches visuelles telles que la segmentation d'images ou la détection d'objets, ainsi qu'à des tâches contenant des données textuelles ou des séries temporelles unidimensionnelles. Dans les deux cas, nous considérons la présence du déséquilibre des ensembles de données comme un problème central et nous proposons des solutions pour en atténuer les effets. Un aperçu des questions abordées dans les différents chapitres ainsi que les contributions correspondantes est donné ci-dessous.

- Notre première contribution, décrite dans le **Chapitre 3**, propose une nouvelle approche, appelée apprentissage actif à un seul stade, pour répondre au problème de démarrage à froid de l'apprentissage actif profond. Comme nous l'avons dit précédemment, le processus d'apprentissage actif itératif a besoin d'un ensemble de données initiales étiquetées, suffisamment grand pour pouvoir être utilisé pour lancer le processus d'apprentissage itératif. En s'inspirant de l'apprentissage par transfert et de l'adaptation au domaine, nous proposons d'utiliser une représentation à usage général qui est apprise sur un domaine source. Cette proposition est en accord avec d'autres schémas efficaces d'apprentissage d'étiquettes, et en particulier avec le schéma d'apprentissage few-shot dans lequel il a été démontré que des résultats de pointe peuvent être obtenus par une bonne représentation apprise [216]. Le

principe de notre approche est d'utiliser une représentation apprise sur un grand **ensemble de données source étiquetées pour représenter les échantillons** et de sélectionner, en fonction de leurs représentations, un ensemble diversifié d'échantillons à présenter pour annotation. Notre approche suppose également que l'ensemble de données non étiquetées peut être déséquilibré et que notre approche peut limiter ce déséquilibre.

- Nous avons ensuite proposé d'améliorer le cadre classique de l'apprentissage actif itératif qui suppose, contrairement à la contribution précédente, un sous-ensemble initial étiqueté suffisant du domaine cible pour répondre à deux de ses limites. Dans le **Chapitre 4**, nous nous concentrons sur une meilleure gestion du déséquilibre du jeu de données. Nous proposons une nouvelle stratégie de sélection qui **priorise les classes minoritaires** pour une sélection équilibrée et informative. Nous comparons également les méthodes de la première contribution qui reposent sur un domaine source avec les méthodes développées dans le cadre itératif.

  le chapitre 5, nous proposons une nouvelle stratégie pour combiner les objectifs d'information et de représentativité dans la sélection. Nous introduisons une nouvelle fonction d'acquisition qui sélectionne les échantillons en fonction des estimations des modèles appris dans les itérations actuelles et précédentes. Les échantillons pour lesquels il y a un décalage maximal vers l'incertitude entre les deux dernières prédictions des modèles appris sont favorisés. Le choix est fait de sélectionner les échantillons pour lesquels le modèle est le plus susceptible d'oublier et donc de trouver difficile l'apprentissage.

- Notre dernière contribution traite du cas où de nouvelles classes peuvent être vues dans les données de production (i.e. domaine dynamique) et s'appuie sur l'apprentissage incrémentiel. Notre travail se concentre sur la limitation des oublis catastrophiques tout en prenant en compte le déséquilibre des classes. Nous proposons une étude détaillée de l'apprentissage incrémentiel déséquilibré en nous concentrant sur les méthodes de calibration dont l'objectif est de réduire le biais de prédiction entre les classes majoritaires et minoritaires. Nous proposons également deux nouvelles méthodes de calibration et comparons leurs performances à celles des méthodes existantes. Cette partie de notre travail est présentée dans le **Chapitre 6**.

Nos travaux ont été publiés dans des conférences et revues internationales récentes.

- **Chapitre 3** Aggarwal Umang, Adrian Popescu, et Céline Hudelot. "Apprentissage actif pour les ensembles de données déséquilibrés". Actes de la conférence d'hiver IEEE/CVF sur les applications de la vision par ordinateur. 2020.
- **Chapitre 4** Aggarwal Umang, Adrian Popescu, et Céline Hudelot. "Mino-

rity Class Oriented Active Learning for Imbalanced Datasets". 2020 25e Conférence internationale sur la reconnaissance des formes (ICPR). IEEE, 2021.

- **Chapitre 5** Aggarwal Umang, Adrian Popescu, et Céline Hudelot. "Optimizing Active Learning for Low Annotation Budgets" arxiv 2201.07200 in cs.CV .
- **Chapitre 6** Aggarwal Umang, Adrian Popescu, Eden Belouadah, et Céline Hudelot. "Une étude comparative des méthodes de calibration pour l'apprentissage incrémentiel en classe déséquilibrée". Outils et applications multimédia (2021) : 1-20.

Bibliographie

[1] Naoki Abe. “Query learning strategies using boosting and bagging”. In : *Proc. of 15 ˆ< th> Int. Cmf. on Machine Learning (ICML98)* (1998), p. 1-9.

[2] Umang Aggarwal, Adrian Popescu et Celine Hudelot. “Active Learning for Imbalanced Datasets”. In : *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Mars 2020.

[3] Hongjoon Ahn et al. “Uncertainty-based continual learning with adaptive regularization”. In : *arXiv preprint arXiv :1905.11614* (2019).

[4] Rahaf Aljundi, Punarjay Chakravarty et Tinne Tuytelaars. “Expert Gate : Lifelong Learning with a Network of Experts”. In : *Conference on Computer Vision and Pattern Recognition*. CVPR. 2017.

[5] Bang An, Wenjun Wu et Huimin Han. “Deep active learning for text classification”. In : *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*. 2018, p. 1-6.

[6] Dana Z Anderson. *Neural Information Processing Systems : Proceedings of a conference held in Denver, Colorado, November 1987*. Springer Science & Business Media, 1988.

[7] Jordan T Ash et al. “Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds.” In : *ICLR*. 2020.

[8] C Atkinson et al. “Pseudo-Recursal : Solving the Catastrophic Forgetting Problem in Deep Neural Networks”. In : *arXiv preprint arXiv :1802.03875* (2018).

[9] Josh Attenberg et Seyda Ertekin. “Class imbalance and active learning”. In : *Imbalanced Learning : Foundations, Algorithms, and Applications* (2013), p. 101-149.

[10] Philip Bachman, Alessandro Sordoni et Adam Trischler. “Learning algorithms for active learning”. In : *arXiv preprint arXiv :1708.00088* (2017).

[11] L Douglas Baker et Andrew Kachites McCallum. “Distributional clustering of words for text classification”. In : *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 1998, p. 96-103.

[12] Colin Bellinger, Christopher Drummond et Nathalie Japkowicz. “Beyond the Boundaries of SMOTE - A Framework for Manifold-Based Synthetically Oversampling”. In : *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I*. 2016, p. 248-263.

[13] Eden Belouadah, Adrian Popescu et Ioannis Kanellos. “A comprehensive study of class incremental learning algorithms for visual tasks”. In : *Neural Networks* (2020).

[14] Eden Belouadah et al. “Active Class Incremental Learning for Imbalanced Datasets”. In : *European Conference on Computer Vision*. Springer. 2020, p. 146-162.

[15] William H. Beluch et al. “The Power of Ensembles for Active Learning in Image Classification”. In : *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 2018, p. 9368-9377.

[16] Yoshua Bengio, Aaron C. Courville et Pascal Vincent. “Unsupervised Feature Learning and Deep Learning : A Review and New Perspectives”. In : *CoRR* abs/1206.5538 (2012). arXiv : 1206.5538. url : http://arxiv.org/abs/1206.5538.

[17] Yoshua Bengio et al. “An Empirical Investigation of Catastrophic Forgeting in Gradient-Based Neural Networks”. In : (2013).

[18] David Berthelot et al. “Mixmatch : A holistic approach to semi-supervised learning”. In : *Advances in Neural Information Processing Systems*. 2019, p. 5049-5059.

[19] Aditya R Bhattacharya, Ji Liu et Shayok Chakraborty. “A Generic Active Learning Framework for Class Imbalance Applications.” In : *BMVC*. 2019, p. 121.

[20] Zalán Bodó, Zsolt Minier et Lehel Csató. “Active learning with clustering”. In : *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*. JMLR Workshop et Conference Proceedings. 2011, p. 127-139.

[21] Lukas Bossard, Matthieu Guillaumin et Luc Van Gool. “Food-101 – Mining Discriminative Components with Random Forests”. In : *European Conference on Computer Vision*. 2014.

[22] Andrew J Bremner, David J Lewkowicz et Charles Spence. *Multisensory development*. Oxford University Press, 2012.

[23] Mateusz Buda, Atsuto Maki et Maciej A. Mazurowski. “A systematic study of the class imbalance problem in convolutional neural networks”. In : *Neural Networks* 106 (2018), p. 249-259. doi : 10.1016/j.neunet.2018.07.011. url : https://doi.org/10.1016/j.neunet.2018.07.011.

[24] Qiong Cao et al. “VGGFace2 : A Dataset for Recognising Faces across Pose and Age”. In : *13th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2018, Xi'an, China, May 15-19, 2018*. 2018, p. 67-74. doi : 10.1109/FG.2018.00020. url : https://doi.org/10.1109/FG.2018.00020.

[25] Francisco M. Castro et al. “End-to-End Incremental Learning”. In : *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*. 2018, p. 241-257. doi : 10.1007/978-3-030-01258-8_15. url : https://doi.org/10.1007/978-3-030-01258-8%5C_15.

[26] Gert Cauwenberghs et Tomaso Poggio. “Incremental and decremental support vector machine learning”. In : *Advances in neural information processing systems*. 2001, p. 409-415.

[27] Shayok Chakraborty et al. “Active Batch Selection via Convex Relaxations with Guaranteed Solution Bounds”. In : *IEEE Trans. Pattern Anal. Mach. Intell.* 37.10 (2015), p. 1945-1958. doi : 10.1109/TPAMI.2015.2389848. url : https://doi.org/10.1109/TPAMI.2015.2389848.

[28] Haw-Shiuan Chang, Erik Learned-Miller et Andrew McCallum. “Active bias : Training more accurate neural networks by emphasizing high variance samples”. In : *Advances in Neural Information Processing Systems*. 2017, p. 1002-1012.

[29] Nitesh V. Chawla et al. “SMOTE : Synthetic Minority Over-sampling Technique”. In : *J. Artif. Intell. Res.* 16 (2002), p. 321-357. doi : 10.1613/jair.953. url : https://doi.org/10.1613/jair.953.

[30] Zhiyuan Chen et Bing Liu. “Lifelong machine learning”. In : *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12.3 (2018), p. 1-207.

[31] Ekin D Cubuk et al. “Autoaugment : Learning augmentation policies from data”. In : *arXiv preprint arXiv :1805.09501* (2018).

[32] Aron Culotta et Andrew McCallum. “Reducing Labeling Effort for Structured Prediction Tasks”. In : *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*. 2005, p. 746-751.

[33] Mark Culp et George Michailidis. “Graph-based semisupervised learning”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.1 (2007), p. 174-179.

[34] Wei Dai et Daniel Berleant. “Benchmarking contemporary deep learning hardware and frameworks : A survey of qualitative metrics”. In : *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*. IEEE. 2019, p. 148-155.

[35] Sanjoy Dasgupta et Daniel Hsu. “Hierarchical sampling for active learning”. In : *Proceedings of the 25th international conference on Machine learning*. 2008, p. 208-215.

[36] Sanjoy Dasgupta et Daniel J. Hsu. “Hierarchical sampling for active learning”. In : *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*. 2008, p. 208-215.

[37] Matthias Delange et al. “A continual learning survey : Defying forgetting in classification tasks”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[38] Jia Deng et al. “ImageNet : A large-scale hierarchical image database”. In : *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. 2009, p. 248-255.

[39] Li Deng. “A tutorial survey of architectures, algorithms, and applications for deep learning”. In : *APSIPA Transactions on Signal and Information Processing* 3 (2014).

[40] Jacob Devlin et al. “Bert : Pre-training of deep bidirectional transformers for language understanding”. In : *arXiv preprint arXiv :1810.04805* (2018).

[41] Terrance DeVries et Graham W Taylor. “Dataset augmentation in feature space”. In : *arXiv preprint arXiv :1702.05538* (2017).

[42] Happiness Ugochi Dike et al. “Unsupervised learning based on artificial neural network : A review”. In : *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*. IEEE. 2018, p. 322-327.

[43] Gregory Ditzler et al. “Learning in nonstationary environments : A survey”. In : *IEEE Computational Intelligence Magazine* 10.4 (2015), p. 12-25.

[44] Carl Doersch, Abhinav Gupta et Alexei A Efros. “Unsupervised visual representation learning by context prediction”. In : *Proceedings of the IEEE international conference on computer vision*. 2015, p. 1422-1430.

[45] Jeff Donahue et Karen Simonyan. “Large scale adversarial representation learning”. In : *arXiv preprint arXiv :1907.02544* (2019).

[46] Mengnan Du, Ninghao Liu et Xia Hu. “Techniques for interpretable machine learning”. In : *Communications of the ACM* 63.1 (2019), p. 68-77.

[47] Parijat Dube et al. “Automatic Labeling of Data for Transfer Learning”. In : *nature* 192255 (2019), p. 241.

[48] Melanie Ducoffe et Frederic Precioso. “Adversarial active learning for deep networks : a margin based approach”. In : *arXiv preprint arXiv :1802.09841* (2018).

[49] Charles Elkan. “The Foundations of Cost-Sensitive Learning”. In : *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001*. 2001, p. 973-978.

[50] Seyda Ertekin, Jian Huang et C Lee Giles. “Active learning for class imbalance problem”. In : *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 2007, p. 823-824.

[51] Fan Feng et al. “Challenges in task incremental learning for assistive robotics”. In : *IEEE Access* 8 (2019), p. 3434-3441.

[52] Ugo Fiore et al. “Using generative adversarial networks for improving classification effectiveness in credit card fraud detection”. In : *Information Sciences* 479 (2019), p. 448-455.

[53] Pierre Foret et al. “Sharpness-Aware Minimization for Efficiently Improving Generalization”. In : *arXiv preprint arXiv :2010.01412* (2020).

[54] Yoav Freund et Robert E Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In : *Journal of computer and system sciences* 55.1 (1997), p. 119-139.

[55] JuiHsi Fu et SingLing Lee. “Certainty-based active learning for sampling imbalanced datasets”. In : *Neurocomputing* 119 (2013), p. 350-358.

[56] Yifan Fu, Xingquan Zhu et Bin Li. “A survey on instance selection for active learning”. In : *Knowledge and information systems* 35.2 (2013), p. 249-283.

[57] Yarin Gal, Riashat Islam et Zoubin Ghahramani. “Deep Bayesian Active Learning with Image Data”. In : *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, p. 1183-1192.

[58] Mingfei Gao et al. “Consistency-Based Semi-Supervised Active Learning : Towards Minimizing Labeling Cost”. In : *arXiv preprint arXiv :1910.07153* (2019).

[59] Mingfei Gao et al. “Consistency-based semi-supervised active learning : Towards minimizing labeling cost”. In : *European Conference on Computer Vision*. Springer. 2020, p. 510-526.

[60] Leon A Gatys, Alexander S Ecker et Matthias Bethge. “A neural algorithm of artistic style”. In : *arXiv preprint arXiv :1508.06576* (2015).

[61] Yonatan Geifman, Guy Uziel et Ran El-Yaniv. “Bias-reduced uncertainty estimation for deep neural classifiers”. In : *arXiv preprint arXiv :1805.08206* (2018).

[62] Yonatan Geifman et Ran El-Yaniv. “Deep active learning over the long tail”. In : *arXiv preprint arXiv :1711.00941* (2017).

[63] Alexander Gepperth et Barbara Hammer. “Incremental learning algorithms and applications”. In : *European symposium on artificial neural networks (ESANN)*. 2016.

[64] Bhavya Ghai et al. “Explainable Active Learning (XAL) Toward AI Explanations as Interfaces for Machine Teachers”. In : *Proceedings of the ACM on Human-Computer Interaction* 4.CSCW3 (2021), p. 1-28.

[65] Spyros Gidaris, Praveer Singh et Nikos Komodakis. “Unsupervised representation learning by predicting image rotations”. In : *arXiv preprint arXiv :1803.07728* (2018).

[66] Ross Girshick. “Fast r-cnn”. In : *Proceedings of the IEEE international conference on computer vision*. 2015, p. 1440-1448.

[67] Daniel Gissin et Shai Shalev-Shwartz. “Discriminative active learning”. In : *arXiv preprint arXiv :1907.06347* (2019).

[68] Ben Goertzel. “Artificial general intelligence : concept, state of the art, and future prospects”. In : *Journal of Artificial General Intelligence* 5.1 (2014), p. 1-48.

[69] Ian Goodfellow et al. “Generative adversarial nets”. In : *Advances in neural information processing systems* 27 (2014).

[70] Ian J Goodfellow, Jonathon Shlens et Christian Szegedy. “Explaining and harnessing adversarial examples”. In : *arXiv preprint arXiv :1412.6572* (2014).

[71] Denis Gudovskiy et al. “Deep Active Learning for Biased Datasets via Fisher Kernel Self-Supervision”. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2020.

[72] David Gunning et al. “XAI—Explainable artificial intelligence”. In : *Science Robotics* 4.37 (2019).

[73] Chuan Guo et al. “On Calibration of Modern Neural Networks”. In : *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, p. 1321-1330.

[74] Chuan Guo et al. “On calibration of modern neural networks”. In : *International Conference on Machine Learning*. PMLR. 2017, p. 1321-1330.

[75] Haixiang Guo et al. “Learning from class-imbalanced data : Review of methods and applications”. In : *Expert Syst. Appl.* 73 (2017), p. 220-239. doi : 10.1016/j.eswa.2016.12.035. url : https://doi.org/10.1016/j.eswa.2016.12.035.

[76] Lan-Zhe Guo et al. “Safe deep semi-supervised learning for unseen-class unlabeled data”. In : *International Conference on Machine Learning*. PMLR. 2020, p. 3897-3906.

[77] Barbara Hammer, Haibo He et Thomas Martinetz. “Learning and modeling big data”. In : *ESANN*. 2014.

[78] Hui Han, Wenyuan Wang et Binghuan Mao. “Borderline-SMOTE : A New Over-Sampling Method in Imbalanced Data Sets Learning”. In : *Advances in Intelligent Computing, International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I*. 2005, p. 878-887.

[79] Lei Han et al. “Local uncertainty sampling for large-scale multi-class logistic regression”. In : *arXiv preprint arXiv :1604.08098* (2016).

[80] Degan Hao et al. “Inaccurate labels in weakly-supervised deep learning : Automatic identification and correction and their impact on classification performance”. In : *IEEE journal of biomedical and health informatics* 24.9 (2020), p. 2701-2710.

[81] Demis Hassabis et al. “Neuroscience-inspired artificial intelligence”. In : *Neuron* 95.2 (2017), p. 245-258.

[82] Hamed Hassanzadeh et MohammadReza Keyvanpour. “A machine learning based analytical framework for semantic annotation requirements”. In : *arXiv preprint arXiv :1104.4950* (2011).

[83] Elmar Haussmann et al. “Scalable active learning for object detection”. In : *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, p. 1430-1435.

[84] Mohammad Havaei et al. “Brain tumor segmentation with deep neural networks”. In : *Medical image analysis* 35 (2017), p. 18-31.

[85] Haibo He et Edwardo A Garcia. “Learning from imbalanced data”. In : *IEEE Transactions on knowledge and data engineering* 21.9 (2009), p. 1263-1284.

[86] Haibo He et Edwardo A. Garcia. “Learning from Imbalanced Data”. In : *IEEE Trans. Knowl. Data Eng.* 21.9 (2009), p. 1263-1284.

[87] Jingrui He et Jaime G Carbonell. “Nearest-neighbor-based active learning for rare category detection”. In : *Advances in neural information processing systems*. 2008, p. 633-640.

[88] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In : *Conference on Computer Vision and Pattern Recognition*. CVPR. 2016.

[89] Kaiming He et al. “Deep residual learning for image recognition”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 770-778.

[90] Kaiming He et al. “Momentum contrast for unsupervised visual representation learning”. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, p. 9729-9738.

[91] Tao He et al. “Towards better uncertainty sampling : Active learning with multiple views for deep convolutional neural network”. In : *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2019, p. 1360-1365.

[92] Paulina Hensman et David Masko. “The impact of imbalanced training data for convolutional neural networks”. In : *Degree Project in Computer Science, KTH Royal Institute of Technology* (2015).

[93] Geoffrey E. Hinton, Oriol Vinyals et Jeffrey Dean. “Distilling the Knowledge in a Neural Network”. In : *CoRR* abs/1503.02531 (2015). arXiv : 1503.02531. url : http://arxiv.org/abs/1503.02531.

[94] Dorit S Hochbaum et David B Shmoys. “A best possible heuristic for the k-center problem”. In : *Mathematics of operations research* 10.2 (1985), p. 180-184.

[95] Thorsten Hoeser et Claudia Kuenzer. “Object detection and image segmentation with deep learning on Earth observation data : A review-part I : Evolution and recent trends”. In : *Remote Sensing* 12.10 (2020), p. 1667.

[96] MD Hossain et al. “A comprehensive survey of deep learning for image captioning”. In : *ACM Computing Surveys (CSUR)* 51.6 (2019), p. 118.

[97] Saihui Hou et al. “Learning a Unified Classifier Incrementally via Rebalancing”. In : *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. 2019, p. 831-839.

[98] Neil Houlsby et al. “Bayesian active learning for classification and preference learning”. In : *arXiv preprint arXiv :1112.5745* (2011).

[99] Wei-Ning Hsu et Hsuan-Tien Lin. “Active learning by learning”. In : *Twenty-Ninth AAAI conference on artificial intelligence*. Citeseer. 2015.

[100] Peiyun Hu et al. “Active Learning with Partial Feedback”. In : *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019.

[101] Chen Huang et al. “Learning Deep Representation for Imbalanced Classification”. In : *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, p. 5375-5384. doi : 10.1109/CVPR.2016.580. url : https://doi.org/10.1109/CVPR.2016.580.

[102] Sheng-Jun Huang, Rong Jin et Zhi-Hua Zhou. “Active Learning by Querying Informative and Representative Examples”. In : *IEEE Trans. Pattern Anal. Mach. Intell.* 36.10 (2014), p. 1936-1949.

[103] Shamsul Huda et al. “A hybrid feature selection with ensemble classification for imbalanced healthcare data : A case study for brain tumor diagnosis”. In : *IEEE access* 4 (2016), p. 9145-9154.

[104] Bettina Hüttenrauch. “Limitations of data augmentation and outlook”. In : *Targeting Using Augmented Data in Database Marketing*. Springer, 2016, p. 279-290.

[105] Dino Ienco et al. “Clustering based active learning for evolving data streams”. In : *International Conference on Discovery Science*. Springer. 2013, p. 79-93.

[106] Prateek Jain et Ashish Kapoor. “Active learning for large multi-class problems”. In : *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, p. 762-769.

[107] Ashish Jaiswal et al. “A survey on contrastive self-supervised learning”. In : *Technologies* 9.1 (2021), p. 2.

[108] Nathalie Japkowicz et Shaju Stephen. “The class imbalance problem : A systematic study”. In : *Intell. Data Anal.* 6.5 (2002), p. 429-449. url : http://content.iospress.com/articles/intelligent-data-analysis/ida00103.

[109] Khurram Javed et Faisal Shafait. “Revisiting Distillation and Incremental Classifier Learning”. In : *CoRR* abs/1807.02802 (2018). arXiv : 1807.02802. url : http://arxiv.org/abs/1807.02802.

[110] George F. Jenks. *Optimal data classification for choropleth maps*. T. 2. University of Kansas Department of Geography Occasional Paper, 1977.

[111] Longlong Jing et Yingli Tian. “Self-supervised visual feature learning with deep neural networks : A survey”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[112] Elmer H. Johnson. “Elementary Applied Statistics : For Students in Behavioral Science”. In : *Social Forces* 44.3 (mars 1966), p. 455-456. issn : 0037-7732. doi : 10.1093/sf/44.3.455. eprint : http://oup.prod.sis.lan/sf/article-pdf/44/3/455/6507689/44-3-455.pdf. url : https://doi.org/10.1093/sf/44.3.455.

[113] Justin M Johnson et Taghi M Khoshgoftaar. “Survey on deep learning with class imbalance”. In : *Journal of Big Data* 6.1 (2019), p. 1-54.

[114] Dongmin Kang et al. “Confidence Calibration for Incremental Learning”. In : *IEEE Access* 8 (2020), p. 126648-126660.

[115] Guoliang Kang et al. “Patchshuffle regularization”. In : *arXiv preprint arXiv :1707.07103* (2017).

[116] Ashraf Khalil et al. “Investigating Bias in Facial Analysis Systems : A Systematic Review”. In : *IEEE Access* 8 (2020), p. 130751-130761.

[117] Salman H. Khan et al. “Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data”. In : *IEEE Trans. Neural Netw. Learning Syst.* 29.8 (2018), p. 3573-3587. doi : 10.1109/TNNLS.2017.2732482. url : https://doi.org/10.1109/TNNLS.2017.2732482.

[118] Diederik P Kingma et al. “Semi-supervised learning with deep generative models”. In : *arXiv preprint arXiv :1406.5298* (2014).

[119] Ksenia Konyushkova, Raphael Sznitman et Pascal Fua. “Learning active learning from data”. In : *Advances in Neural Information Processing Systems*. 2017, p. 4225-4235.

[120] Simon Kornblith, Jonathon Shlens et Quoc V. Le. “Do Better ImageNet Models Transfer Better ?” In : *CoRR* abs/1805.08974 (2018).

[121] Bartosz Krawczyk. “Learning from imbalanced data : open challenges and future directions”. In : *Progress in Artificial Intelligence* 5.4 (2016), p. 221-232.

[122] Jan Kremer, Kim Steenstrup Pedersen et Christian Igel. “Active learning with support vector machines”. In : *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery* 4.4 (2014), p. 313-326.

[123] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Rapp. tech. 2009.

[124] Alex Krizhevsky, Geoffrey Hinton et al. “Learning multiple layers of features from tiny images”. In : (2009).

[125] Alex Krizhevsky, Ilya Sutskever et Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In : *Advances in neural information processing systems*. 2012, p. 1097-1105.

[126] Miroslav Kubat et Stan Matwin. “Addressing the Curse of Imbalanced Training Sets : One-Sided Selection”. In : *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997*. 1997, p. 179-186.

[127] Punit Kumar et Atul Gupta. “Active learning query strategies for classification, regression, and clustering : a survey”. In : *Journal of Computer Science and Technology* 35.4 (2020), p. 913-945.

[128] Alina Kuznetsova et al. “The open images dataset v4”. In : *International Journal of Computer Vision* (2020), p. 1-26.

[129] Agostina J Larrazabal et al. “Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis”. In : *Proceedings of the National Academy of Sciences* 117.23 (2020), p. 12592-12594.

[130] Gustav Larsson, Michael Maire et Gregory Shakhnarovich. “Colorization as a proxy task for visual understanding”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, p. 6874-6883.

[131] Yann LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In : *Neural computation* 1.4 (1989), p. 541-551.

[132] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In : *Proceedings of the IEEE* 86.11 (1998), p. 2278-2324.

[133] Hansang Lee, Minseok Park et Junmo Kim. “Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning”. In : *2016 IEEE international conference on image processing (ICIP)*. IEEE. 2016, p. 3713-3717.

[134] Christiane Lemke, Marcin Budka et Bogdan Gabrys. “Metalearning : a survey of trends and technologies”. In : *Artificial intelligence review* 44.1 (2015), p. 117-130.

[135] Der-Chiang Li, Chiao-Wen Liu et Susan C Hu. “A learning method for the class imbalance problem with medical data sets”. In : *Computers in biology and medicine* 40.5 (2010), p. 509-518.

[136] Fei-Fei Li, Andrej Karpathy et Justin Johnson. “CS231n : Convolutional Neural Networks for Visual Recognition 2016”. In : (). url : http://cs231n.stanford.edu/.

[137] Mingkun Li et Ishwar K Sethi. “Confidence-based active learning”. In : *IEEE transactions on pattern analysis and machine intelligence* 28.8 (2006), p. 1251-1261.

[138] Shuangtao Li et al. “Learning more robust features with adversarial training”. In : *arXiv preprint arXiv :1804.07757* (2018).

[139] Xianglin Li, Runqiu Guo et Jun Cheng. “Incorporating Incremental and Active Learning for Scene Classification”. In : *11th International Conference on Machine Learning and Applications, ICMLA, Boca Raton, FL, USA, December 12-15, 2012. Volume 1*. 2012, p. 256-261.

[140] Zhizhong Li et Derek Hoiem. “Learning Without Forgetting”. In : *European Conference on Computer Vision*. ECCV. 2016.

[141] Tsung-Yi Lin et al. “Focal loss for dense object detection”. In : *Proceedings of the IEEE international conference on computer vision*. 2017, p. 2980-2988.

[142] Tsung-Yi Lin et al. “Microsoft coco : Common objects in context”. In : *European conference on computer vision*. Springer. 2014, p. 740-755.

[143] Zachary C Lipton. “The Mythos of Model Interpretability : In machine learning, the concept of interpretability is both important and slippery.” In : *Queue* 16.3 (2018), p. 31-57.

[144] Bo Liu et al. “Feature space transfer for data augmentation”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, p. 9090-9098.

[145] Li Liu et al. “Deep Learning for Generic Object Detection : A Survey”. In : *International Journal of Computer Vision* 128 (fév. 2020). doi : 10.1007/s11263-019-01247-4.

[146] Xiao Liu et al. “Self-supervised learning : Generative or contrastive”. In : *arXiv preprint arXiv :2006.08218* 1.2 (2020).

[147] Xu-Ying Liu, Jianxin Wu et Zhi-Hua Zhou. “Exploratory Undersampling for Class-Imbalance Learning”. In : *IEEE Trans. Systems, Man, and Cybernetics, Part B* 39.2 (2009), p. 539-550. doi : 10.1109/TSMCB.2008.2007853. url : https://doi.org/10.1109/TSMCB.2008.2007853.

[148] Ze Liu et al. “Swin Transformer : Hierarchical Vision Transformer using Shifted Windows”. In : *arXiv preprint arXiv :2103.14030* (2021).

[149] Nan Lu et al. “Mitigating Overfitting in Supervised Classification from Two Unlabeled Datasets : A Consistent Risk Correction Approach”. In : *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Sous la dir. de Silvia Chiappa et Roberto Calandra. T. 108. Proceedings of Machine Learning Research. PMLR, 26–28 Aug 2020, p. 1115-1125. url : http://proceedings.mlr.press/v108/lu20c.html.

[150] Tomasz Maciejewski et Jerzy Stefanowski. “Local neighbourhood extension of SMOTE for mining imbalanced data”. In : *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2011, part of the IEEE Symposium Series on Computational Intelligence 2011, April 11-15, 2011, Paris, France*. 2011, p. 104-111.

[151] Maciej A Mazurowski et al. “Training neural network classifiers for medical decision making : The effects of imbalanced datasets on classification performance”. In : *Neural networks* 21.2-3 (2008), p. 427-436.

[152] Andrew Kachites McCallumzy et Kamal Nigamy. “Employing EM and pool-based active learning for text classification”. In : *Proc. International Conference on Machine Learning (ICML)*. Citeseer. 1998, p. 359-367.

[153] Michael McCloskey et Neal J Cohen. “Catastrophic interference in connectionist networks : The sequential learning problem”. In : *Psychology of learning and motivation*. T. 24. Elsevier, 1989, p. 109-165.

[154] Michael Mccloskey et Neil J. Cohen. “Catastrophic Interference in Connectionist Networks : The Sequential Learning Problem”. In : *The Psychology of Learning and Motivation* 24 (1989), p. 104-169.

[155] Thomas Mensink et al. “Distance-Based Image Classification : Generalizing to New Classes at Near-Zero Cost”. In : *IEEE Trans. Pattern Anal. Mach. Intell.* 35.11 (2013), p. 2624-2637.

[156] Martial Mermillod, Aurélia Bugaiska et Patrick Bonin. “The stability-plasticity dilemma : Investigating the continuum from catastrophic forgetting to age-limited learning effects”. In : *Frontiers in psychology* 4 (2013), p. 504.

[157] George A Miller. “WordNet : a lexical database for English”. In : *Communications of the ACM* 38.11 (1995), p. 39-41.

[158] Ishan Misra et Laurens van der Maaten. “Self-supervised learning of pretext-invariant representations”. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, p. 6707-6717.

[159] Shinichi Mizokami. “Deep active learning from the perspective of active learning theory”. In : *Deep active learning*. Springer, 2018, p. 79-91.

[160] Robert Munro Monarch. *Human-in-the-Loop Machine Learning : Active learning and annotation for human-centered AI*. Simon et Schuster, 2021.

[161] Robert Moskovitch et al. “Improving the detection of unknown computer worms activity using active learning”. In : *Annual Conference on Artificial Intelligence*. Springer. 2007, p. 489-493.

[162] T Nathan Mundhenk, Daniel Ho et Barry Y Chen. “Improvements to context based self-supervised learning”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 9339-9348.

[163] Ion Muslea, Steven Minton et Craig A Knoblock. “Selective sampling with redundant views”. In : *AAAI/IAAI*. 2000, p. 621-626.

[164] Cuong V Nguyen et al. “Variational continual learning”. In : *arXiv preprint arXiv :1710.10628* (2017).

[165] Alexandru Niculescu-Mizil et Rich Caruana. “Predicting good probabilities with supervised learning”. In : *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*. 2005, p. 625-632. doi : 10.1145/1102351.1102430. url : https://doi.org/10.1145/1102351.1102430.

[166] Hyeonwoo Noh et al. “Large-Scale Image Retrieval with Attentive Deep Local Features”. In : *Proc. ICCV*. 2017. url : https://arxiv.org/abs/1612.06321.

[167] Hyeonwoo Noh et al. “Large-Scale Image Retrieval with Attentive Deep Local Features”. In : *ICCV*. IEEE Computer Society, 2017, p. 3476-3485.

[168] Augustus Odena. “Semi-supervised learning with generative adversarial networks”. In : *arXiv preprint arXiv :1606.01583* (2016).

[169] Avital Oliver et al. “Realistic evaluation of deep semi-supervised learning algorithms”. In : *arXiv preprint arXiv :1804.09170* (2018).

[170] Maxime Oquab et al. “Learning and transferring mid-level image representations using convolutional neural networks”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, p. 1717-1724.

[171] Yassine Ouali, Céline Hudelot et Myriam Tami. “An Overview of Deep Semi-Supervised Learning”. In : *arXiv preprint arXiv :2006.05278* (2020).

[172] John D Owens et al. “GPU computing”. In : *Proceedings of the IEEE* 96.5 (2008), p. 879-899.

[173] Kunkun Pang et al. “Meta-learning transferable active learning policies by deep reinforcement learning”. In : *arXiv preprint arXiv :1806.04798* (2018).

[174] German I Parisi et al. “Continual lifelong learning with neural networks : A review”. In : *Neural Networks* 113 (2019), p. 54-71.

[175] German Ignacio Parisi et al. “Continual Lifelong Learning with Neural Networks : A Review”. In : *Neural Networks* 113 (2019).

[176] Adam Paszke et al. “Automatic differentiation in PyTorch”. In : *Advances in Neural Information Processing Systems Workshops*. NIPS-W. 2017.

[177] Swarnajyoti Patra et Lorenzo Bruzzone. “A batch-mode active learning technique based on multiple uncertainty for SVM classifier”. In : *IEEE Geoscience and Remote Sensing Letters* 9.3 (2011), p. 497-501.

[178] Fabian Pedregosa et al. “Scikit-learn : Machine Learning in Python”. In : *CoRR* abs/1201.0490 (2012). arXiv : 1201 . 0490. url : http://arxiv.org/abs/1201.0490.

[179] Benedikt Pfülb et Alexander Gepperth. “A comprehensive, application-oriented study of catastrophic forgetting in dnns”. In : *arXiv preprint arXiv :1905.08101* (2019).

[180] John Platt et al. “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods”. In : *Advances in large margin classifiers* 10.3 (1999), p. 61-74.

[181] V Jothi Prakash et Dr LM Nithya. “A survey on semi-supervised learning techniques”. In : *arXiv preprint arXiv :1402.4645* (2014).

[182] Ariadna Quattoni et Antonio Torralba. “Recognizing indoor scenes”. In : *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. 2009, p. 413-420.

[183] Joaquin Quionero-Candela et al. *Dataset Shift in Machine Learning*. The MIT Press, 2009. isbn : 0262170051, 9780262170055.

[184] M Mostafizur Rahman et DN Davis. “Addressing the class imbalance problem in medical datasets”. In : *International Journal of Machine Learning and Computing* 3.2 (2013), p. 224.

[185] Ali Sharif Razavian et al. “CNN Features Off-the-Shelf : An Astounding Baseline for Recognition”. In : *Conference on Computer Vision and Pattern Recognition Workshop*. CVPR-W. 2014.

[186] Muhammad Imran Razzak, Saeeda Naz et Ahmad Zaib. “Deep learning for medical image processing : Overview, challenges and the future”. In : *Classification in BioApps* (2018), p. 323-350.

[187] Sylvestre-Alvise Rebuffi, Hakan Bilen et Andrea Vedaldi. “Learning multiple visual domains with residual adapters”. In : *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 2017, p. 506-516.

[188] Sylvestre-Alvise Rebuffi et al. “iCaRL : Incremental Classifier and Representation Learning”. In : *Conference on Computer Vision and Pattern Recognition*. CVPR. 2017.

[189] Pengzhen Ren et al. “A survey of deep active learning”. In : *arXiv preprint arXiv :2009.00236* (2020).

[190] Michael D. Richard et Richard P. Lippmann. “Neural Network Classifiers Estimate Bayesian *a posteriori* Probabilities”. In : *Neural Computation* 3.4 (1991), p. 461-483. doi : 10.1162/neco.1991.3.4.461. url : https://doi.org/10.1162/neco.1991.3.4.461.

[191] Soumya Roy, Asim Unmesh et Vinay P Namboodiri. “Deep active learning for object detection.” In : *BMVC*. T. 362. 2018, p. 91.

[192] Sebastian Ruder et Barbara Plank. “Strong baselines for neural semi-supervised learning under domain shift”. In : *arXiv preprint arXiv :1804.09530* (2018).

[193] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In : *International Journal of Computer Vision* 115.3 (2015), p. 211-252. doi : 10.1007/s11263-015-0816-y. url : https://doi.org/10.1007/s11263-015-0816-y.

[194] Olga Russakovsky et al. “Imagenet large scale visual recognition challenge”. In : *International journal of computer vision* 115.3 (2015), p. 211-252.

[195] Andrei A. Rusu et al. “Progressive Neural Networks”. In : *CoRR* abs/1606.04671 (2016). arXiv : 1606.04671. url : http://arxiv.org/abs/1606.04671.

[196] Tobias Scheffer, Christian Decomain et Stefan Wrobel. “Mining the Web with Active Hidden Markov Models”. In : *Proceedings of the 2001 IEEE International Conference on Data Mining, 29 November - 2 December 2001, San Jose, California, USA*. 2001, p. 645-646.

[197] Christopher Schröder et Andreas Niekler. “A survey of active learning for text classification using deep neural networks”. In : *arXiv preprint arXiv :2008.07267* (2020).

[198] Ozan Sener et Silvio Savarese. “Active Learning for Convolutional Neural Networks : A Core-Set Approach”. In : *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 2018.

[199] Burr Settles. *Active learning literature survey*. Rapp. tech. University of Winsconsin, 2010.

[200] Burr Settles et Mark Craven. “An Analysis of Active Learning Strategies for Sequence Labeling Tasks”. In : *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. 2008.

[201] Burr Settles, Mark Craven et Soumya Ray. “Multiple-instance active learning”. In : *Advances in neural information processing systems* 20 (2007), p. 1289-1296.

[202] Claude Elwood Shannon. “A Mathematical Theory of Communication”. In : 27.3 (juill. 1948), p. 379-423. url : https://ieeexplore.ieee.org/document/6773024/.

[203] Victor S Sheng, Foster Provost et Panagiotis G Ipeirotis. “Get another label? improving data quality and data mining using multiple, noisy labelers”. In : *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008, p. 614-622.

[204] Hanul Shin et al. “Continual learning with deep generative replay”. In : *arXiv preprint arXiv :1705.08690* (2017).

[205] Connor Shorten et Taghi M Khoshgoftaar. “A survey on image data augmentation for deep learning”. In : *Journal of Big Data* 6.1 (2019), p. 1-48.

[206] Karen Simonyan et Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In : *arXiv preprint arXiv :1409.1556* (2014).

[207] Samarth Sinha, Sayna Ebrahimi et Trevor Darrell. “Variational Adversarial Active Learning”. In : *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[208] Arnold WM Smeulders et al. “Content-based image retrieval at the end of the early years”. In : *IEEE Transactions on pattern analysis and machine intelligence* 22.12 (2000), p. 1349-1380.

[209] Kihyuk Sohn et al. “Fixmatch : Simplifying semi-supervised learning with consistency and confidence”. In : *arXiv preprint arXiv :2001.07685* (2020).

[210] Amarnag Subramanya et Partha Pratim Talukdar. “Graph-based semi-supervised learning”. In : *Synthesis Lectures on Artificial Intelligence and Machine Learning* 8.4 (2014), p. 1-125.

[211] Chen Sun et al. “Revisiting unreasonable effectiveness of data in deep lea rning era”. In : *Proceedings of the IEEE international conference on computer vision*. 2017, p. 843-852.

[212] Youssef Tamaazousti et al. “Learning More Universal Representations for Transfer-Learning”. In : *arXiv :1712.09708* (2017).

[213] Youssef Tamaazousti et al. “Learning more universal representations for transfer-learning”. In : *IEEE transactions on pattern analysis and machine intelligence* (2019).

[214] Luke Taylor et Geoff Nitschke. “Improving deep learning with generic data augmentation”. In : *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2018, p. 1542-1547.

[215] Stefano Teso. “Toward Faithful Explanatory Active Learning with Self-explainable Neural Nets”. In : *Proceedings of the Workshop on Interactive Adaptive Learning (IAL 2019)*. CEUR Workshop Proceedings. 2019, p. 4-16.

[216] Yonglong Tian et al. “Rethinking few-shot image classification : a good embedding is all you need ?” In : *Computer Vision–ECCV 2020 : 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer. 2020, p. 266-282.

[217] Tatiana Tommasi et al. “A deeper look at dataset bias”. In : *Domain adaptation in computer vision applications*. Springer, 2017, p. 37-55.

[218] Simon Tong et Edward Chang. “Support vector machine active learning for image retrieval”. In : *Proceedings of the ninth ACM international conference on Multimedia*. 2001, p. 107-118.

[219] Antonio Torralba, Alexei A Efros et al. “Unbiased look at dataset bias.” In : *CVPR*. T. 1. 2. Citeseer. 2011, p. 7.

[220] Toan Tran et al. “Bayesian generative active deep learning”. In : *International Conference on Machine Learning*. PMLR. 2019, p. 6295-6304.

[221] Selen Uguroglu et Jaime Carbonell. “Feature selection for transfer learning”. In : *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2011, p. 430-442.

[222] Muhammad Usama et al. “Unsupervised machine learning for networking : Techniques, applications and research challenges”. In : *IEEE Access* 7 (2019), p. 65579-65615.

[223] Jesper E Van Engelen et Holger H Hoos. “A survey on semi-supervised learning”. In : *Machine Learning* 109.2 (2020), p. 373-440.

[224] Wouter Van Gansbeke et al. “Scan : Learning to classify images without labels”. In : *European Conference on Computer Vision*. Springer. 2020, p. 268-285.

[225] Ragav Venkatesan et al. “A Strategy for an Uncompromising Incremental Learner”. In : *CoRR* abs/1705.00744 (2017).

[226] Hui Wang et al. “Dynamic Pseudo-Label Generation for Weakly Supervised Object Detection in Remote Sensing Images”. In : *Remote Sensing* 13.8 (2021), p. 1461.

[227] Keze Wang et al. “Cost-effective active learning for deep image classification”. In : *IEEE Transactions on Circuits and Systems for Video Technology* 27.12 (2016), p. 2591-2600.

[228] Mei Wang et Weihong Deng. “Deep visual domain adaptation : A survey”. In : *Neurocomputing* 312 (2018), p. 135-153.

[229] Shoujin Wang et al. “Training deep neural networks on imbalanced data sets”. In : *2016 international joint conference on neural networks (IJCNN)*. IEEE. 2016, p. 4368-4374.

[230] Xinyue Wang et al. “Important sampling based active learning for imbalance classification”. In : *Science China Information Sciences* 63.8 (2020), p. 1-14.

[231] Yaqing Wang et al. “Generalizing from a few examples : A survey on few-shot learning”. In : *ACM Computing Surveys (CSUR)* 53.3 (2020), p. 1-34.

[232] Yu Emma Wang, Gu-Yeon Wei et David Brooks. “Benchmarking tpu, gpu, and cpu platforms for deep learning”. In : *arXiv preprint arXiv :1907.10701* (2019).

[233] Yu-Xiong Wang, Deva Ramanan et Martial Hebert. “Growing a Brain : Fine-Tuning by Increasing Model Capacity”. In : *Conference on Computer Vision and Pattern Recognition*. CVPR. 2017.

[234] Kapil Keshao Wankhade, Kalpana C Jondhale et Vijaya R Thool. “A hybrid approach for classification of rare class data”. In : *Knowledge and Information Systems* 56.1 (2018), p. 197-221.

[235] Kai Wei, Rishabh Iyer et Jeff Bilmes. “Submodularity in data subset selection and active learning”. In : *International Conference on Machine Learning*. 2015, p. 1954-1963.

[236] Karl Weiss, Taghi M Khoshgoftaar et DingDing Wang. “A survey of transfer learning”. In : *Journal of Big data* 3.1 (2016), p. 1-40.

[237] Yi Wu et al. “Sampling strategies for active learning in personal photo retrieval”. In : *2006 IEEE International Conference on Multimedia and Expo*. IEEE. 2006, p. 529-532.

[238] Yue Wu et al. “Large Scale Incremental Learning”. In : *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. 2019, p. 374-382.

[239] Zhirong Wu et al. “Unsupervised feature learning via non-parametric instance discrimination”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 3733-3742.

[240] Qiang Yang et Xindong Wu. “10 challenging problems in data mining research”. In : *International Journal of Information Technology & Decision Making* 5.04 (2006), p. 597-604.

[241] Donggeun Yoo et In So Kweon. “Learning Loss for Active Learning”. In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2019.

[242] Jaehong Yoon et al. “Lifelong learning with dynamically expandable networks”. In : *arXiv preprint arXiv :1708.01547* (2017).

[243] Michelle Yuan, Hsuan-Tien Lin et Jordan Boyd-Graber. “Cold-start active learning through self-supervised language modeling”. In : *arXiv preprint arXiv :2010.09535* (2020).

[244] Bianca Zadrozny et Charles Elkan. “Transforming classifier scores into accurate multiclass probability estimates”. In : *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*. 2002, p. 694-699. doi : 10.1145/775047.775151. url : https://doi.org/10.1145/775047.775151.

[245] Friedemann Zenke, Ben Poole et Surya Ganguli. “Continual learning through synaptic intelligence”. In : *International Conference on Machine Learning*. PMLR. 2017, p. 3987-3995.

[246] Chen Zeno et al. “Task agnostic continual learning using online variational bayes”. In : *arXiv preprint arXiv :1803.10123* (2018).

[247] Beichen Zhang et al. “State-Relabeling Adversarial Active Learning”. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2020.

[248] Chuanhai Zhang et al. “Similarity-based active learning for image classification under class imbalance”. In : *2018 IEEE international conference on data mining (ICDM)*. IEEE. 2018, p. 1422-1427.

[249] Kaipeng Zhang et al. “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks”. In : *IEEE Signal Process. Lett.* 23.10 (2016), p. 1499-1503. doi : 10.1109/LSP.2016.2603342. url : https://doi.org/10.1109/LSP.2016.2603342.

[250] Man Zhang et al. “A survey of semi-and weakly supervised semantic segmentation of images”. In : *Artificial Intelligence Review* 53.6 (2020), p. 4259-4288.

[251] Richard Zhang, Phillip Isola et Alexei A Efros. “Colorful image colorization”. In : *European conference on computer vision*. Springer. 2016, p. 649-666.

[252] Fedor Zhdanov. “Diverse mini-batch active learning”. In : *arXiv preprint arXiv :1901.05954* (2019).

[253] Bolei Zhou et al. “Places : A 10 million image database for scene recognition”. In : *IEEE transactions on pattern analysis and machine intelligence* 40.6 (2017), p. 1452-1464.

[254] Zhi-Hua Zhou. “A brief introduction to weakly supervised learning”. In : *National science review* 5.1 (2018), p. 44-53.

[255] Zhi-Hua Zhou et Xu-Ying Liu. “Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem”. In : *IEEE Trans. Knowl. Data Eng.* 18.1 (2006), p. 63-77. doi : 10.1109/TKDE.2006.17. url : https://doi.org/10.1109/TKDE.2006.17.

[256] Zongwei Zhou et al. “Fine-Tuning Convolutional Neural Networks for Biomedical Image Analysis : Actively and Incrementally”. In : *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, p. 4761-4772.

[257] Jia-Jie Zhu et José Bento. “Generative adversarial active learning”. In : *arXiv preprint arXiv :1702.07956* (2017).

[258] Jingbo Zhu et Eduard Hovy. “Active learning for word sense disambiguation with methods for addressing the class imbalance problem”. In : *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. 2007, p. 783-790.

[259] Xiaojin Zhu, John Lafferty et Zoubin Ghahramani. “Combining active learning and semi-supervised learning using gaussian fields and harmonic functions”. In : *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*. T. 3. 2003.

[260] Fuzhen Zhuang et al. “A comprehensive survey on transfer learning”. In : *Proceedings of the IEEE* 109.1 (2020), p. 43-76.

[261] Maciej Zieba et Jakub M Tomczak. “Boosted SVM with active learning strategy for imbalanced data”. In : *Soft Computing* 19.12 (2015), p. 3357-3368.