Active Learning for Imbalanced Datasets

Umang Aggarwal[1,2], Adrian Popescu[1], Céline Hudelot[2]
(1) Université Paris-Saclay, CEA, Département Intelligence Ambiante et Systèmes Interactifs
(2) Université Paris-Saclay, CentraleSupélec, Mathématiques et Informatique pour la Complexité et les Systèmes
91191 Gif-sur-Yvette, France
umang.aggarwal,adrian.popescu@cea.fr,celine.hudelot@centralesupelec.fr

Abstract

Active learning increases the effectiveness of labeling when only subsets of unlabeled datasets can be processed manually. To our knowledge, existing algorithms are designed under the assumption that datasets are balanced. However, many real-life datasets are actually imbalanced and we propose two adaptations of active learning to tackle imbalance. First, we modify acquisition functions to select samples by taking advantage of a deep model pretrained on a source domain. Second, we introduce a balancing step in the acquisition process to reduce the imbalance of the labeled subset. Evaluation is done with four imbalanced datasets using existing active learning methods and their modifications introduced here. Results show that our adaptations are useful as long as knowledge from the source domain is transferable to target domains.

1. Introduction

The availability of large annotated datasets is a central requirement to train robust deep learning models. Although large scale image collections are now available, for instance on the Web, their manual labeling remains time-consuming. Active learning (AL) algorithms [30] are designed to select representative subsets of unlabeled datasets for manual labeling and thus reduce the cost of the annotation process. The topic has recently re-emerged in the context of deep learning [1, 8, 11, 21, 29, 33, 35]. However, to our knowledge, none of the proposed approaches tackles the problem of dataset imbalance, which affects wide array of datasets built for real-life applications. Imbalance also appears for research datasets, even if they are built with strong supervision. ImageNet [7] and Open Images [16] (built for object recognition and detection), Google Landmarks [19] (tourist landmarks recognition) or MS-CELEB-1M [9] (face recognition) are all imbalanced. For instance, classes from the full ImageNet dataset are represented by 648 images on average with a standard deviation of 527. In contrast, the ILSVRC subset of ImageNet [27], widely used in the community, is nearly perfectly balanced.

In AL, an acquisition function is used to select samples for manual labeling, given an annotation budget. Most classical AL works [30] assume that access is provided to a manually labeled subset which includes all classes of the unlabeled dataset at the start of the process. This assumption is made to kick start the AL procedure by first training a model on the labeled subset and then using this model to select more images from the unlabeled dataset. Then, the model can be updated until the whole budget is spent. An important drawback of this classical scenario is that the acquisition functions select images based on features learned on a small subset, which might be weak or unstable [30]. This problem is stringent for deep representations which are data-intensive by nature.

Inspired by more recent works [1, 36, 8], we assume to have access to a deep model pretrained on a source domain but no access to a seed labeled subset. Also inspired by these approaches, we work with a one shot active learning scenario. Images are selected by acquisition functions based on features provided by the pretrained model. The AL models are only created after labeling the full subset allowed by the AL budget. This scenario is more realistic than the classical one because it does not suppose that the total number of classes is known in advance. It is also interesting because an iterative training of deep model to include each new labeled sample would be time consuming. The existence of a pretrained model is also a realistic hypothesis which is extensively exploited in transfer learning [15, 25].

Learning from imbalanced datasets leads to a prediction bias towards majority classes over minority classes for classical machine learning algorithms [10, 13]. A similar conclusion was recently presented in [3], where the authors study the effect of data imbalance on deep learning algorithms. Our hypothesis is that existing active learning approaches tend to reproduce or even worsen the majority bias. This bias is particularly important for the acquisition of samples for manual labeling, the key AL step which should find a representative subset of the unlabeled dataset.

The first contribution is to modify acquisition functions in order to adapt them to the one-shot scenario. To do this, we exploit two related outputs of a deep model trained on a source domain to select useful samples for labeling. Top-1 predictions of source classes are used to privilege samples for which the source model is highly confident. Then, a memory stores source classes which are associated to samples which were already labeled in order to obtain a semantically diversified subset. A new sample is labeled only if its associated source class was not already encountered.

The second contribution is to introduce a sample balancing step which reduces the propagation of imbalance of 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. During balancing, priority is given to classes which are underrepresented among the samples that were already labeled. Note that the switch between the two acquisition modes is transparent for the human annotator and the required labeling effort is identical.

Evaluation is done with imbalanced versions of four public datasets designed for different visual tasks and three AL labeling budgets. The modified acquisition functions are compared to their original formulation, to random selection and to $core-set$ [29], a recent geometric-based approach, with and without the balancing step. Results indicate that both the modified acquisition functions and sample balancing are useful for three out of four. An analysis of source knowledge transferability is provided to explain why the proposed approach does not work for the fourth dataset.

2. Related Work

Active learning is a well-studied problem in classical machine learning literature and has recently been investigated in a deep learning context. Our contribution is to examine the fitness of existing methods for imbalanced datasets and to propose adaptations for such datasets.

An overview of classical AL approaches is provided in [30]. A first class of methods based on informativeness exploits the uncertainty of the classifier predictions to select informative examples. The most common measures to estimate the uncertainty of samples are based on least confidence first [5], margin sampling [28] or entropy [32]. In all cases, the least certain samples which are often at the borders between classes are favored. However, while they can help in improving the decision boundary of the classifier, they are not representative for the data distribution as a whole [30]. A second class of methods is based on density and was proposed to improve the representativeness of the selected samples. Information density is an early example of such strategy [31], while k-centers and core sets were tested more recently [29]. Algorithms were also devised to combine sample informativeness and representativeness. K-means and hierarchical clustering based approaches were explored in [18] and [6]. QUIRE [12] is an example of such algorithm which deploys a min-max view of AL. In [4], entropy and KL-divergence are combined to obtain uncertain and representative examples. None of these functions were explicitly tested with imbalanced datasets and our hypothesis is that adaptations of them are needed in this context. We focus on functions which can be modified with our approach.

Recent works on active learning include a deep learning component. The authors of [21] use a flow of belief over a graph version of the unlabeled dataset to select samples which minimize joint entropy of nodes and select a non-uniform number of samples per AL batch. In [11], the manual labeling effort is analyzed and labels are progressively pruned as the process advances in order to simplify it. Very recently, an algorithm which learns a loss function specifically for AL was proposed in [35]. Inspired by uncertainty-based functions, samples which are likely to produce wrong predictions are suggested for annotation. Monte Carlo (MC) dropout [8] exploits the softmax predictions of a deep model with random dropout masks to generate to model uncertainty. An ensemble approach which combines multiple snapshots of the same network training was proposed in [1]. Coupled with a variation ratio function [14], 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. They can be applied to any acquisition function and we provide results with ensembles in the supplementary material. The works cited above usually need a labeled seed set of samples to initialize the AL process. With low budgets, the initial deep learning models are likely to be suboptimal or even not trainable. As an alternative, we take inspiration from works in transfer learning [25, 15] and exploit a model pretrained on a source domain.

Active incremental fine-tuning (AIFT) [36] proposes to use iterative fine tuning in order to improve AL for biomedical 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 select relevant examples. AIFT 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, (2) the acquisition functions used to select AL candidate samples and (3) the criterion used to select candidate samples.

Also relevant are works from imbalanced learning [3], which we use to test their impact on results.

3. Active Learning for Imbalanced Datasets

3.1. Problem Description

We propose a formalization of AL in an imbalanced setting. 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 class labels $\{y_1, ..., y_N\}$ of the source domain and n_S the number of annotated instances. We now consider a target domain $\mathcal{D}_T$ only represented by an unlabeled dataset $\mathbb{D}_T^U : x_i \in \mathcal{X}$ for $i = 1..n_T$ for a target domain distribution $\mathbb{P}_T$ with n_T the number of samples in the target domain. The objective of Active Learning is to select the best subset $\mathbb{D}_T^L$ of $\mathbb{D}_T^U$ of cardinal b (the budget) for manual labeling in order to maximize the performance of its associated model over the test set $\mathbb{D}_T^t$. We also consider that $\mathbb{D}_T^U$ 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 samples per class. The higher the ratio between σ and μ is, the stronger the imbalance of the dataset will be.

The proposed AL scenario encompasses three steps. First, a deep model $\mathcal{M}_S$ is learned over the source domain and includes two main components. The first is a feature extractor $\mathcal{F}_S : x_i \rightarrow \mathbb{R}^d$, with d the size of the feature vector. The second is a classifier followed by a soft-max function $\mathcal{P}_S : \mathbb{R}^d \rightarrow P(Y_S)$ which outputs the probability distribution over the N classes of the source domain. The two components of the model $\mathcal{M}_S$ are used to extract features f and predictions $p(y_s)$ for all samples x^i from $\mathcal{D}_T^U$.

Second, a labeled dataset $\mathbb{D}_T^L$ is obtained via the application of an acquisition function $\mathcal{AF}$ [30] which raises two challenges. 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 [24]. 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 AL budgets. We introduce: (1) adaptations of uncertainty-based $\mathcal{AF}$ by diversifying samples based on source dataset predictions and (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.

Third, a model $\mathcal{M}_T$ is trained over $\mathbb{D}_T^L$ to test AL performance. 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. Finally, we apply thresholding, which uses the prior class probabilities to augment the scores of minority classes and is shown to out-perform a large array of data sampling and classifier level methods for object recognition using deep learning models [3]. Optionally, weakly supervised learning could 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.

3.2. Acquisition Functions

As discussed in Section 2, a wealth of AL acquisition functions were proposed which focus either on uncertainty, representativeness or a combination of them. We briefly describe the most representative $\mathcal{AF}$ as reported in recent papers [1, 8, 29]. In our AL scenario, no manual annotation of the target dataset is available at the start of the process and uncertainty measures are computed using the outputs of the source model $\mathcal{M}_S$. Their usefulness is thus subject to the degree of representation transferability between the source and the target domains.

3.2.1 Uncertainty-based Functions

Uncertainty-based methods allow an AL method to query the instances which lie close to the decision boundary of the model. In deep AL contexts [1], these methods exploit the classifier predictions obtained with the pretrained model $\mathcal{M}_S$. The hypothesis is that these instances are the ones on which the model is most likely to be uncertain and hence the most informative to fine-tune the decision boundary. Several uncertainty measures have been proposed in literature [30] and we selected the most influential ones here, namely: entropy, margin sampling, least confidence.

Entropy Sampling is a concept borrowed from information theory and is based on the global shape of class predictions of $\mathcal{M}_S$. It is defined as:

$$ent = \texttt{invsort}_{\forall x \in \mathcal{D}_T^U}(H(x)) \tag{1}$$

with: `invsort` a function which sorts samples x in decreasing order based on $H(x)$, the entropy of x as calculated over $\mathcal{M}_S$ predictions. This $\mathcal{AF}$ method selects samples with highest entropy from the top of the list ent provided by Equation 1.

Margin Sampling is an effective active learning method which computes the uncertainty of an instance x by comparing its top 2 predictions of the model. It is defined as:

$$ms = \texttt{sort}_{\forall x \in \mathcal{D}_T^U}(max(p(y_s)) - max_2(p(y_s))) \tag{2}$$

with: $max(p(y_s))$ and $max_2(p(y_s))$ are the top 2 predicted classes for the sample x and `sort` a sorting function. This

$\mathcal{AF}$ favors samples which minimize the difference between the top two predictions of samples from the list ms. Note that ms can be seen as a truncated form of ent, computed only over the top predictions of each sample.

Least Confidence Sampling is another uncertainty-based approach which selects instances on which the model $\mathcal{M}_S$ is least confident, i.e. favors the lowest probabilities available. It is defined as:

$$lc = sort_{\forall x \in \mathcal{D}_T^U}(max(p(y_s))) \tag{3}$$

Values from the lc list are sorted from lowest to maximum predicted value from the set of top-1 softmax probabilities $p(y_s)$ for each sample x in $\mathcal{D}_T^U$.

The presented uncertainty-based $\mathcal{AF}$ only exploit the class predictions provided by the pretrained model $\mathcal{M}_S$. They are thus likely to be suboptimal for imbalanced dataset because, due to their definition, majority classes might be favored and the initial imbalance of the unlabeled datasets might be reinforced. As a result, the overall efficiency of active learning will be reduced.

3.2.2 Diversified Certainty-based Functions

As discussed in Section 2, acquisition functions can be designed to select informative or representative samples or even a combination of both [12]. To take full advantage of the initial model, we propose a sample diversification strategy based on $\mathcal{M}_S$ top predictions. This strategy operates under the assumption that, due to 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 uncertainty-based methods. The $\mathcal{AF}$ presented in Equations 1, 2, 3 are first inverted so as to put the most certain examples at the top of the sampling lists. The final form of the functions is thus:

$$ent_{inv}^{div} = div(\texttt{sort}_{\forall x \in \mathcal{D}_T^U}(H(x))) \tag{4}$$

$$ms_{inv}^{div} = div(\texttt{invsort}_{\forall x \in \mathcal{D}_T^U}(max(p(y_s)) - max_2(p(y_s)))) \tag{5}$$

$$lc_{inv}^{div} = div(\texttt{invsort}_{\forall x \in \mathcal{D}_T^U}(max(p(y_s)))) \tag{6}$$

, with: div a diversification function discussed below.

Inversion is necessary to sort the samples according to certainty. div assigns every unlabeled sample to its predicted source class. The selection of samples is performed by iterating over the source classes, selecting one example per source class, till the budget is filled. The underlining assumption is that the samples assigned to different source classes with high certainty would be different from each other. Further, a diverse set of images is selected by giving equal representation to samples from all the source classes.

3.2.3 Geometric-based Functions

Geometric approaches are based on building a subset using the feature extractor $\mathcal{F}_S$ of the pretrained model $\mathcal{M}_S$ on the unlabeled dataset. One recent method [29] creates a $core-set$ of the unlabeled dataset by solving the greedy k-center problem. It tries to minimize the distance between any unlabeled point in the unlabeled target dataset to its closest labeled point in the source dataset. Hence at every step, it selects the point which is at a maximum distance from its closest labeled point. We implement this method by randomly selecting the first labeled point and then solving Equation 7:

$$\max_{\forall x_u \in \mathcal{D}_T^U} \min_{x_l \in \mathcal{D}_T^L} d(f(x_u), f(x_l)) \tag{7}$$

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 $\mathbb{D}_T^U$. Note that we tried to apply the diversification procedure to the $coreset$ too but results were inconclusive. This negative finding is probably explained by the fact that geometric-based functions live in the feature space, and diversification is applied to the classifier predictions.

3.3. Active Learning with Balancing

The switch from classical AL to balancing step needs to be done in order 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. It is activated using the following rule:

$$b - m <= c_{ur} \times (\mu(or) - \mu(ur)) \tag{8}$$

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 in $\mathbb{D}_T^L$ (under- and over-representation are defined w.r.t. the mean number of class samples after labeling m images).

For every m value, Equation 8 tests if there are enough samples left until b to fill in the gap between the samples of under-represented and over-represented classes. 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) ur classes simply might not have enough samples in $\mathbb{D}_T^U$ and (2) some of the samples attributed during balancing will be directed towards other classes than intended. The stronger the imbalance of a dataset, the earlier the switch proposed in Equation 8 will be activated. Note that balancing in Equation 8 is designed for imbalanced datasets but it might also affect AL for balanced datasets. An evaluation with this latter setting is provided in the supplementary material

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

$$x_{next} = \min_{\forall i \in \{1, n-m\}} (\frac{d(\mu(F_S(C_{ur}^{min}), F_S(x_i))}{\max_{\forall j \in \{1, c_{or}\}}(d(\mu(F_S(C_j)), F_S(x_i))}) \tag{9}$$

with x_i any of the unlabeled $n-m$ samples at moment m; $d(.,.)$ - the 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. 9 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. We also tested with a version of Eq. 9 in which the denominator was not used and obtained lower performance.

3.4. 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 [25] 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 shown to be particularly prone to imbalance and provide prediction scores biased towards majority classes [3]. 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 test both shallow classifiers and fine tuned models for each fold. Accuracy is averaged over all folds and the strategy which has better performance is selected. Raw performance for the two strategies is provided in the supplementary material.

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 1. Dataset statistics. ir is the imbalance ratio

4. Experiments

4.1. Datasets

The proposed methods are evaluated on four imbalanced datasets and we consider ILSVRC [27] as source domain. We induce imbalance in the publicly available Food-101 [2] (fine-grained food recognition), CIFAR-100 [17] (object recognition), MIT Indoor-67 [23] (indoor scene recognition). In addition, we create IMN-100 a subset of randomly selected 100 leaf classes from ImageNet which are not present in ILSVRC. This last dataset is created to test transfer among classes from the same large collection of images. An 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 1. Imbalance is similar across datasets to facilitate comparability of results. More details about imbalance induction are given in the supplementary material.

4.2. Implementation Details

The Pytorch [20] 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 deeper models might not converge and (2) the number of experiments to run is large and a relatively quick training is needed. A classical fine-tuning strategy is applied when CNNs are used to create $\mathcal{M}_T$ over the labeled subset $\mathbb{D}_T^L$. Parameters of the source training are kept, except for the initial learning rate which is divided by 10. Linear SVMs from scikit-learn [22] are used to create shallow model when transfer learning is used. Their parameters are optimized using 10-fold cross validation over the labeled subset $\mathbb{D}_T^L$. The choice between SVMs and CNNs to create AL models is done by cross-validation, as explained in Subsection 3.4.

4.3. Evaluation Methodology

The size of the budget b is the main criterion used to evaluate the performance of active learning methods [1, 29, 30] and we test $b = \{500, 1000, 2000\}$ for each of them. We present results with existing AL acquisition functions and their modified versions described in Subsection 3.2. Five runs are launched for non-deterministic acquisition functions ($random$ and $core-set$) and their accuracy is averaged to prevent accuracy bias. AL performance is evaluated

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
$core-set$	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 2. Accuracy of the acquisition functions from Subsection 3.2 before balancing. $random$ and $core - set$ 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			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^{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**
ms^{div}_{inv}	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^{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
$core-set$	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. Accuracy of the acquisition functions from Subsection 3.2 after balancing. $random$ and $core - set$ are non deterministic and their performance is averaged over five runs. Best results are presented in bold.

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 used in all experiments is top-1 accuracy. It is calculated as an average over the entire set of classes represented in the test set since the objective is to evaluate the capacity of each AL method to deal with imbalance. The computation includes classes which might not have been discovered during AL. Further discussion of accuracy computation is given in the supplementary material.

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 [26, 34], we propose a global performance score in Equation 10:

$$G_{AL} = \frac{1}{c} \times \sum_{i=1}^{c} \frac{acc_i - acc_{full}}{acc_{max} - acc_{full}} \tag{10}$$

where: c - number of configurations tested; acc_i - top-1 score for each configuration (individual values of each row of Table 2 and Table 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 [3]); 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 [34]. G_{AL} has a negative value and the closer its value to zero, the better the method is. More details about it are provided in the supplementary material.

4.4. Performance of Acquisition Functions

A first important finding provided by Table 2 is that existing $\mathcal{AF}$ are not well adapted for imbalanced datasets. 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 $core - set$ method has global performance equivalent to that of random sampling. Our results confirm the conclusions of [29, 1] regarding the fact that random sampling is a strong AL baseline.

A second important finding is that the proposed $\mathcal{AF}$ adaptations are efficient since performance is improved for

	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
$core-set$	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 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 $core-set$ 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
$core-set$	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 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 $core-set$ because these methods are not deterministic and their performance is averaged over five runs.

all uncertainty-based methods when diversification is applied to their inversed definitions as discussed in Subsection 3.2. The performance gain is particularly interesting for the modified versions of entropy ent_{inv}^{div} and least confidence (lc_{inv}^{div}) which gain 0.57 and 0.45 G_{AL} points compared to $random$. As shown by the intermediate results 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.

The analysis of individual configurations, ent_{inv}^{div} and lc_{inv}^{div} 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. For $b = 2000$, the accuracy of the best methods tends to level because the chances to pick a good subset of the training set are higher and the AL task becomes less relevant. 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 4.6.

In Table 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. Only the main methods from Table 2 are kept. 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. Results are rather well correlated to accuracy, with ent_{inv}^{div} 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 discovery and of imbalance ratio. Interestingly, while $random$ samples are more balanced for MIT-67 compared to ent_{inv}^{div}, accuracy remains better for the latter method. This is probably an effect of the fact that the labeled samples are more representative of each class for

ent_{inv}^{div} compared to a random selection. The results in Table 4 also validate our hypothesis that the application of acquisition functions worsens the global imbalance of $\mathbb{D}_T^U$. 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.

4.5. Influence of Balancing

Balancing provides a consequent improvement for all $\mathcal{AF}$ tested. The G_{AL} scores after balancing (Table 5) are clearly better than those obtained before balancing (Table 4). The G_{AL} score for $random$ moves from -0.792 to -0.653, while that of ent_{inv}^{div} goes from -0.739 to -0.612. lc_{inv}^{div} remains second best but with an increased gap compared to ent_{inv}^{div}. We note also that balancing improves performance of acquisition function for the Food-101 dataset. In particular, ent_{inv}^{div} 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 4 and Table 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_{inv}^{div} 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.

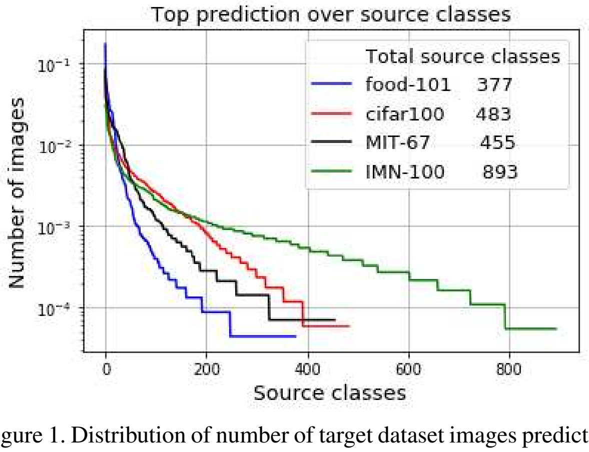

Figure 1. Distribution of number of target dataset images predicted per source class. Source classes are ranked from left to right from most to least frequent. To facilitate comparability, the raw number of predictions is divided by the size of each target dataset. *Best viewed in color.*

4.6. Analysis of Transferability

The distance from source to target domains conditions the success of transfer learning [25]. 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 2 and 3 indicate that the distance between the ILSVRC source is highest for Food-101. We deepen this simple estimation of transferability in Figure 1. 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 1 are directly comparable for Food-101, CIFAR-100, 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 2 and 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.

5. Conclusion

We adapt AL for imbalanced visual datasets by modifying acquisition functions and by introducing a balancing step during manual labeling. The modified acquisition functions take advantage of a pretrained deep model to find representative and diversified samples for manual labeling. 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 pretrained model and the target imbalanced datasets.

Obtained results are encouraging and we will pursue research along three axes. First, we will attempt to propose more elaborate diversification methods for the acquisition functions. Second, we will consider a pretrained model learned on a larger dataset to ensure transferability toward a larger spectrum of target datasets. Finally, we will investigate methods to determine whether representations are transferable between source and target datasets. If this is not the case, it becomes preferable to run random sampling followed by balancing instead of AL acquisition functions.

References

[1] W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler. 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*, pages 9368–9377, 2018.

[2] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

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

[4] S. Chakraborty, V. N. Balasubramanian, Q. Sun, S. Panchanathan, and J. Ye. Active batch selection via convex relaxations with guaranteed solution bounds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(10):1945–1958, 2015.

[5] A. Culotta and A. 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*, pages 746–751, 2005.

[6] S. Dasgupta and D. 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*, pages 208–215, 2008.

[7] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. 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*, pages 248–255, 2009.

[8] Y. Gal, R. Islam, and Z. 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*, pages 1183–1192, 2017.

[9] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, pages 87–102, 2016.

[10] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284, 2009.

[11] P. Hu, Z. C. Lipton, A. Anandkumar, and D. Ramanan. Active learning with partial feedback. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[12] S. Huang, R. Jin, and Z. Zhou. Active learning by querying informative and representative examples. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(10):1936–1949, 2014.

[13] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6(5):429–449, 2002.

[14] E. H. Johnson. Elementary Applied Statistics: For Students in Behavioral Science. *Social Forces*, 44(3):455–456, 03 1966.

[15] S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? *CoRR*, abs/1805.08974, 2018.

[16] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, S. Kamali, M. Malloci, J. Pont-Tuset, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.

[17] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[18] X. Li, R. Guo, and J. 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*, pages 256–261, 2012.

[19] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, pages 3476–3485. IEEE Computer Society, 2017.

[20] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems Workshops*, NIPS-W, 2017.

[21] S. Paul, J. H. Bappy, and A. K. Roy-Chowdhury. Non-uniform subset selection for active learning in structured data. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 830–839, 2017.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. VanderPlas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *CoRR*, abs/1201.0490, 2012.

[23] A. Quattoni and A. 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*, pages 413–420, 2009.

[24] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[25] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *Conference on Computer Vision and Pattern Recognition Workshop*, CVPR-W, 2014.

[26] S. Rebuffi, H. Bilen, and A. 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*, pages 506–516, 2017.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[28] T. Scheffer, C. Decomain, and S. 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, pages 645–646, 2001.

[29] O. Sener and S. 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.

[30] B. Settles. Active learning literature survey. Technical report, University of Winsconsin, 2010.

[31] B. Settles and M. 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.

[32] C. E. Shannon. A mathematical theory of communication. 27(3):379–423, 7 1948.

[33] S. Sharma, A. K. Jha, P. Hegde, and B. Ravindran. Learning to multi-task by active sampling. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[34] Y. Tamaazousti, H. Le Borgne, C. Hudelot, M. E. A. Seddik, and M. Tamaazousti. Learning more universal representations for transfer-learning. *arXiv:1712.09708*, 2017.

[35] D. Yoo and I. S. Kweon. Learning loss for active learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[36] Z. Zhou, J. Y. Shin, L. Zhang, S. R. Gurudu, M. B. Gotway, and J. Liang. 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*, pages 4761–4772, 2017.