Supplementary material: 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

1. Active Learning with Ensembles

The authors of [1] 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 6 and 7 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 in the main paper 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 in the main paper. Note also that $core - set$ with ensemble becomes competitive after the application of balancing in Table 7. It has a global score which is only slightly behind that of our version of diversified entropy.

2. Active Learning with Balanced datasets

The balancing step introduced in Section 3.3 of the paper 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 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 most effective especially at higher budgets and *core-set* the least effective method, as reported in Table 8.

Somewhat surprisingly, results in Table 12 indicate that balancing is beneficial for all acquisition functions. 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

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***
$core - set$	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
$core - set$ $+ens$	*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 6. Accuracy of the acquisition functions with ensemble before balancing. We copy results for the main methods from Table 2 of the main paper. 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 2.

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_{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
ent_{inv}^{div} $+ens$	***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_{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
ls_{inv}^{div} $+ens$	*22.12*	*28.59*	*34.72*	*35.53*	*42.93*	*48.77*	*59.47*	***67.08***	*69.61*	*45.53*	*54.26*	*63.98*	*-0.576*
$core-set$	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
$core-set$ $+ens$	*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 7. Accuracy of the acquisition functions with ensemble after balancing. We copy results for main methods from Table 3 of the main paper. 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.

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
$core-set$	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 8. Accuracy of the acquisition functions with balanced dataset 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	
$random$	**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
$core-set$	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 9. Accuracy of the acquisition functions with balanced dataset after balancing. $random$ and $core-set$ are non deterministic and their performance is averaged over five runs. Best results are presented in bold.

are generally between two and three top-1 accuracy points for lower budgets, which are most interesting in AL since they require the lowest annotation effort. Also interesting, after balancing 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 10 and 11 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 (Tables 4 and 5 of the main paper). This is intuitive since class discovery is simpler when classes are balanced and the odds to find representatives of each class are comparable. The results presented here indicate that the balancing step might be useful for active learning in general and not only for imbalanced datasets. Further evaluation with other acquisition functions and in an iterative setting is needed to reinforce this conclusion.

3. Diversification Pseudo-code

The pseudo code for the diversification procedure described in Subsection 3.2.2 is provided in Algorithm refalg:div. The computational cost of the diversification is negligible since the selection of a new sample to label manually only requires comparison the comparison of its top-1 prediction to a list of top-1 predictions for the samples which were already selected.

	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^{div}_{inv}	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^{div}_{inv}	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
$core-set$	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 10. 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 $core - set$ 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	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^{div}_{inv}	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^{div}_{inv}	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
$core-set$	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 11. 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 $core - set$ because these methods are not deterministic and their performance is averaged over five runs.

4. Dataset Imbalance Induction

As stated in the Section 4.1, a common imbalance induction procedure was applied to all datasets using a target imbalance ratio to guide the pruning process. Similar imbalance ratio was obtained across datasets to facilitate comparability of results. Imbalance induction process for the 4 target dataset was guided to attain similar imbalance ratio to that of the full ImageNet dataset, which is 0.813 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.

5. Evaluation of Accuracy

Performance is calculated by averaging the top-1 accuracy for all N_t classes present in the unlabeled dataset. 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:

$$ACC_t = Acc_f \frac{N_f}{N_t} \tag{1}$$

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.

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
```

Dataset	Food-101 (SVM)			Food-101 (CNN)			CIFAR-100 (SVM)			CIFAR-100 (CNN)		
Budget	500	1000	2000	500	1000	2000	500	1000	2000	500	1000	2000
$random$	**21.74**	**27.62**	**32.54**	**23.02**	**30.63**	**38.68**	27.31	33.66	39.78	16.34	28.50	37.89
ent_{inv}^{div}	20.54	24.55	30.17	19.71	25.60	34.11	32.13	**38.94**	**43.94**	**23.23**	**30.75**	**39.66**
ls_{inv}^{div}	19.09	24.08	29.43	19.13	24.66	33.62	**32.62**	38.46	43.52	22.67	27.70	37.85
$coreSet$	19.30	23.55	28.69	20.07	26.35	34.17	30.04	36.34	42.18	18.77	25.88	32.19
$Full$	42.84			65.85			50.4			59.49		

Table 12. Accuracy of the acquisition functions with unbalanced dataset before balancing for two schemes of training (SVM and CNN). $random$ and $core - set$ are non deterministic and their performance is averaged over five runs. Best results are presented in bold.

6. Comparison of Training Schemes

As mentioned in the main paper Section 3.4, the target model $\mathcal{M}_T$ on the labeled dataset $\mathbb{D}_T^L$ can be trained by transferring deep features from $\mathcal{M}_S$ or by fine-tuning this model. In the first setting a SVM classifier is trained on top of the embedding f from the source model. This setting is advisable when there is strong transferability between the source and target domain, as is established to be the case of Cifar-100. Results from as Table 12 show that for Cifar-100, the SVM setting is significantly better than fine-tuning a CNN, especially for lower budgets. The difference is larger for lower budgets as fine-tuning the model is difficult with less data. In case of Food-101, the features from the source model are not directly usable and it is better to learn all the layers of the model. The upper-bound $Full$ for both the datasets is provided by fine-tuning(CNN), as with enough data, fine-tuning the model over-performs SVM training.