Supplementary material for: Learning Finer-class Networks for Universal Representations

Julien Girard[1][2]
julien.girard2@cea.fr
Youssef Tamaazousti[1][2][3]
youssef.tamaazousti@cea.fr
Hervé Le Borgne[2]
herve.le-borgne@cea.fr
Céline Hudelot[3]
celine.hudelot@centralesupelec.fr

[1] Both authors contributed equally.

[2] CEA LIST
Vision Laboratory,
Gif-sur-Yvette, France.

[3] CentraleSupélec,
MICS Laboratory,
Châtenay-Malabry, France.

Abstract

This document reports some supplementary material that is not required to understand the main article but provide complements or illustrations. Hence, the additional elements were produced using the same version of the approach explained in our main paper and include the following items: (i) the detailed characteristics of the datasets used in this paper (Section 1); (ii) detailed results of the comparison of our method with the baselines (Section 2); and finally (iii) some illustrations of the clusters obtained by the different methods as well as some statistics (Section 3).

1 Datasets: Detailed Characteristics

In Table 1, we report the characteristics of all datasets used in the article to learn CNN on a source-task and to estimate the performances of a universalizing method, that is to say, its performances on a set of target-tasks in the context of transfer-learning. For this, we used the most commonly used dataset as source-task, namely ILSVRC [13] which is a subset of ImageNet [4] that contains 1.2 millions images labeled among 1,000 specific categories. We also follow the literature [16] for fair comparisons and thus used as source-task the ILSVRC* dataset, that corresponds to half of ILSVRC. Regarding the target-tasks, we follow the literature [1, 11, 12, 16] and used ten target-datasets in a classification task. In particular, here we used benchmarks from various domains, namely objects, actions, scenes, as well as, fine-grained objects like aircrafts, birds, cars and plants. In order to show the visual variability of the chosen target-datasets we used to evaluate universalizing methods, we report in Figure 1, some example images of each of them.

2 Comparison to Baselines: Detailed Results

In Figure 3 of the main paper, we reported the synthesis results of the comparison of our methods with several baselines. Thus here, we provide detailed results, that is to say, re-

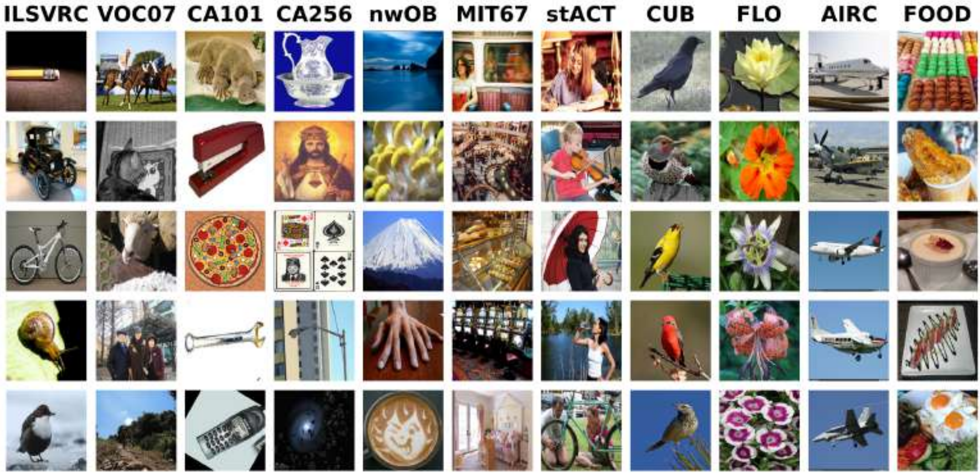

Figure 1: Illustration of some examples of the ILSVRC source-task as well as the ten target-tasks used to evaluate universality. Note the high visual and semantic variability between the different datasets. Best view in PDF.

Datasets	(1)	(2)	(3)	(4)	(5)	(6)	(7)
ILSVRC* [13]	objects	483	1,2K	✗	569,000	48,299	Acc.
ILSVRC [13]	objects	1K	1,2K	✗	1.2M	50,000	Acc.
VOC07 [5]	objects	20	250	✓	5,011	4,952	mAP
NWO [3]	objects	31	700	✓	21,709	14,546	mAP
CA101 [6]	objects	102	30	✗	3,060	3,022	Acc.
CA256 [7]	objects	257	60	✗	15,420	15,187	Acc.
MIT67 [10]	scenes	67	80	✗	5,360	1,340	Acc.
stACT [18]	actions	40	100	✗	4,000	5,532	Acc.
CUB [17]	birds	200	30	✗	5,994	5,794	Acc.
FLO [9]	plants	102	10	✗	1,020	6,149	Acc.
FOOD [2]	food	101	50	✗	5050	5050	Acc.
AIRC [8]	airplanes	100	66	✗	6,667	3,333	Acc.

Table 1: Detailed descriptive of the different datasets used in the article. On top of the table, we describe datasets used as *source-task* and at bottom, those used as *target-task*. For each dataset, we detail seven characteristics. Each column of the table corresponds to a certain characteristic: (1) domain of the images; (2) amount of categories; (3) average amount of training-images per category; (4) whether the dataset contains multiple categories per image (✓) or no (✗); (5) amount of training examples; (6) amount of test examples; and (7) the standard evaluation metric (Accuracy and mean Average Precision, respectively denoted by **Acc.** and **mAP**). Example images of each dataset are presented on Figure 1.

sults of all the methods on each benchmark as well as their average performance on all of them. This is reported in Table 2. Even if already mentioned in the main paper, let recall the most salient results: (i) SpeFiNet is always better than FiNet which is always better than SpeNet, regardless the splitting method; (ii) the proposed BUCBAM splitting method gives better results than the best Kmeans one, at zero cost of parameter cross-validation;

Method	**VOC07**	**CA101**	**CA256**	**NWO**	**MIT67**	**stACT**	**CUB**	**FLO**	**AIRC**	**FOOD**	**Avg.**
	mAP	Acc.	Acc.	mAP	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	
SpeNet (REFERENCE)	66.8	71.1	53.2	52.5	36.0	44.3	36.1	50.5	21.6	25.0	45.7
FiNet Random-K2	66.6	71.8	52.7	52.6	38.9	45.9	34.4	50.9	22.7	24.0	46.0
FiNet Random-K4	66.8	73.1	54.4	52.7	38.9	47.0	34.7	53.5	23.6	25.0	47.0
FiNet Random-K8	67.3	71.9	54.2	51.7	38.7	46.9	34.4	53.6	24.3	24.9	46.8
FiNet Random-K16	66.4	72.4	53.2	51.0	39.7	46.9	35.7	55.9	25.2	26.2	47.3
FiNet Kmeans-K2	66.3	72.2	53.7	52.1	37.7	45.4	33.8	51.5	22.6	24.2	45.9
FiNet Kmeans-K4	66.9	73.0	53.0	51.4	39.2	46.3	35.6	54.3	22.7	25.2	46.8
FiNet Kmeans-K8	66.0	73.2	54.6	50.9	40.7	47.2	36.4	55.6	24.2	26.5	47.5
FiNet Kmeans-K16	64.9	73.8	54.4	50.3	39.0	47.6	36.0	56.8	26.9	26.4	47.6
FiNet Kmeans-K32	63.9	72.1	53.4	48.9	40.2	47.6	36.0	57.8	26.0	26.2	47.2
FiNet Spectral-K16	65.4	72.4	53.7	51.3	39.5	47.4	37.4	55.9	25.3	26.6	47.4
FiNet Affinity	64.5	70.5	52.1	48.5	38.7	45.9	34.8	56.0	25.3	25.5	46.2
FiNet BUCBAM-AS	66.2	72.8	54.3	51.6	39.8	46.7	35.9	56.1	25.3	25.0	47.4
FiNet BUCBAM-SS	65.3	75.4	56.0	48.6	41.6	49.4	37.8	59.8	29.2	28.4	49.2
SpeFiNet Random-K2	**70.0**	76.0	56.8	54.9	41.3	49.4	40.4	57.4	26.4	27.9	50.0
SpeFiNet Random-K4	69.4	76.0	57.9	**55.0**	41.4	49.5	40.0	57.7	27.9	27.6	50.2
SpeFiNet Random-K8	69.8	75.7	57.5	54.6	41.2	50.0	39.8	58.3	27.8	28.6	50.3
SpeFiNet Random-K16	69.4	76.0	55.8	54.3	41.7	49.6	40.3	60.4	28.0	28.9	50.4
SpeFiNet Kmeans-K2	69.2	76.3	57.1	54.7	41.2	49.0	39.0	57.8	26.4	28.3	49.9
SpeFiNet Kmeans-K4	69.6	76.2	57.1	54.2	40.1	49.6	40.8	59.1	26.7	28.3	50.2
SpeFiNet Kmeans-K8	69.1	76.6	57.8	54.1	42.2	49.6	40.4	59.8	28.3	29.3	50.7
SpeFiNet Kmeans-K16	68.6	77.9	58.1	53.9	41.3	50.5	40.8	60.1	29.8	28.4	50.9
SpeFiNet Kmeans-K32	68.4	76.6	57.5	53.5	42.3	50.3	40.4	60.7	28.8	29.0	50.7
SpeFiNet Spectral-K16	68.9	76.0	57.0	54.4	41.0	49.5	41.2	59.3	28.9	29.4	50.5
SpeFiNet Affinity	68.6	75.7	57.1	53.4	41.9	49.1	40.1	59.5	27.6	28.6	50.2
SpeFiNet BUCBAM-AS	69.4	76.3	57.6	54.4	41.1	49.6	40.2	60.1	28.4	28.7	50.6
SpeFiNet BUCBAM-SS	69.1	**78.3**	**59.3**	54.0	**42.7**	**52.0**	**41.8**	**61.7**	**31.4**	**30.8**	**52.1**

Table 2: Comparison of the proposed universalizing methods (BUCBAM) to baselines (Random, Kmeans, Spectral and Affinity) and the reference one (SpeNet). The comparison is carried in a transfer-learning scheme on the ten target-datasets presented in Section 1, for which we report the performances of the methods on each dataset (with standard evaluation-metrics) and the average performance on all the benchmarks (in the last column). All the methods have been learned with the same architecture (AlexNet) on the same initial source-problem (ILSVRC*). As in all the Tables of the main paper, for each dataset, we highlight the score of the best method in bold and those of the second is underlined.

and (iii) the proposed BUCBAM is always better than all other methods, especially Random, Spectral and Affinity. Additionally, in Figure 2, we display the average performances of FiNet-Kmeans-K and SpeFiNet-Kmeans-K according different values of K, which are compared to the performance of a classical SpeNet.

3 Splitting Methods: Statistics and Visualization

In this section we illustrate some interesting properties of the random, cluster and BUCBAM splitting methods. In particular, we first highlight some statistics in Figure 3. Indeed, on top we plot, for each method, the *histogram of amount of images per cluster* for all the specific categories of the initial dataset. Note that, the more the histogram forms a pointed spike, the more the data are balanced. Here we clearly observe that the random splitting method provides the most balanced data, while in contrast other methods tend to contain clusters

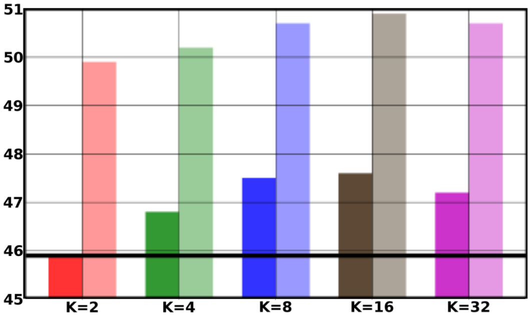

Figure 2: Average performances of FiNet-Kmeans-K and SpeFiNet-Kmeans-K according different values of K. The black line corresponds to the average performance of SpeNet.

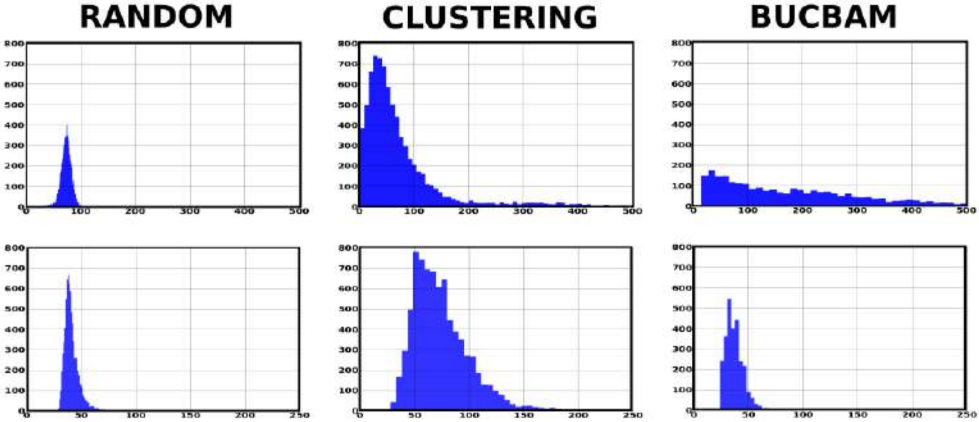

Figure 3: Illustration of some statistics of the different splitting methods, namely random, clustering and BUCBAM. On top, we illustrate the *histogram of the amount of samples per cluster* for all the specific categories of ILSVRC, with the different splitting methods. At bottom, we illustrate the *histogram of the intra-class variance of the cluster*. On vertical axis is the number of cluster.

of various sizes. highly imbalanced data, However, if they provide imbalanced data, it is important to mention that clustering and BUCBAM provide *relevant* clusters that are based on the semantic encoded on the image features, which contain thus samples that are more visually similar. Let also note that, compared to the histogram of clustering that has a long tail and starts at near-zero, the histogram of BUCBAM is more flat and starts around 20, meaning that no tail is modeled in the data and very small clusters are not considered.

At bottom of Figure 3, we reported the *histogram of intra-class variance of the clusters* obtained from all the specific categories, by the three splitting methods. In this, it is important to note that a small width of the histogram means that the set of clusters contains very similar samples. While random provides the smallest width of the peak, it is necessary to observe this is due to the fact that it has almost the same amount of images per cluster, thus it is not relevant. In contrast, BUCBAM that provides a large set of amount of categories, also provides a width of the peak that is lowest than clustering, meaning that it provides clusters with more similar samples.

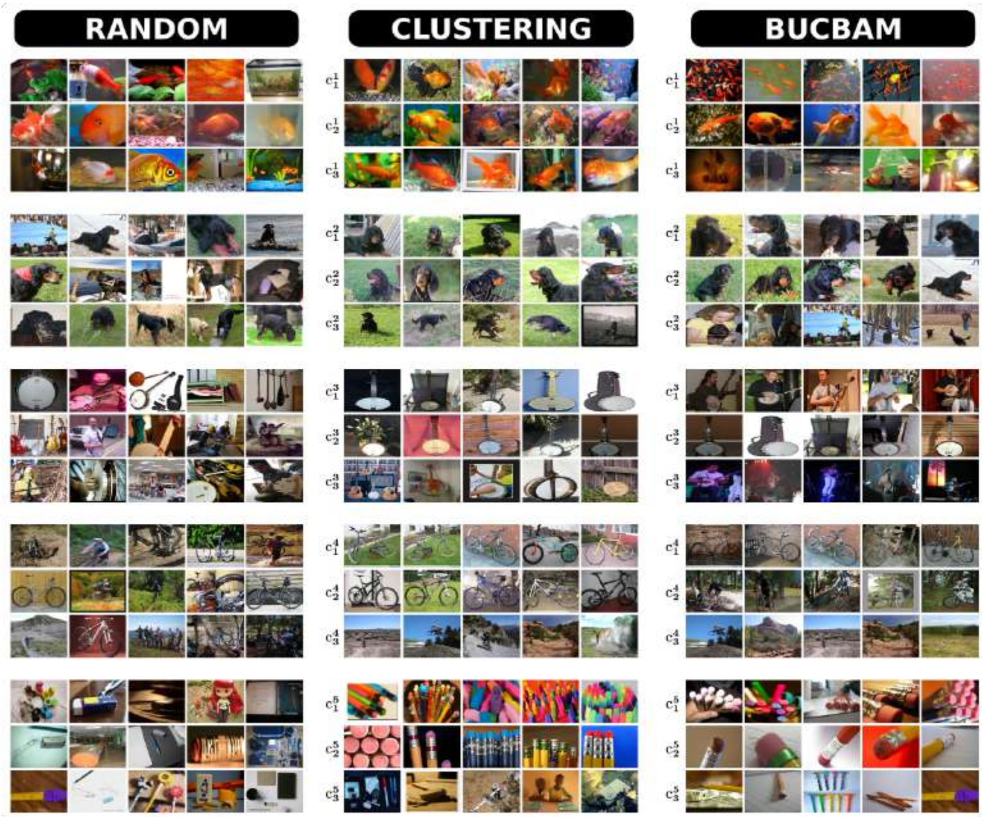

Figure 4: Illustration of the finer categories obtained from five specific categories (five row blocks) with the different methods: random split (left), Kmeans clustering with K=16 (middle) and our BUCBAM proposal (right). In each of the five block, each line shows the five most representative images of a cluster at the new finest-level. Best view in PDF.

While previously, we reported some global statistics of the resulting clusters from the different splitting methods, here we rather show the most representative samples of the clusters obtained by each splitting method. Indeed, this is reported on Figure 4, on which we highlight three clusters (three rows of images) for five specific categories (five blocks of three rows of images). On the left, the clusters are determined from a random distribution within the full specific category, leading to clusters that contain its full diversity. Note by the way, how diverse the specific categories are, and imagine how generic categories (used in [14, 15]) could be, which may explain why the GenNet of [15] may not provide good results (since it is hard from it to discover relevant features). On the contrary, with our splitting method and more precisely the K-means clustering (middle), the clusters exhibits a more coherent aspect. For example, for the *goldfish* category, the c_3^1 cluster report close-up views of fish that are rather seen on their profile. We have a similar behaviour for the *bicycle* category with cluster c_2^4 and c_3^4. With the method we propose (right), the clusters are even more specific than in the K-means case. For instance, for the *goldfish* category, we clearly identify a cluster that represents "many golfishes" (c_1^1), "on goldfish in a close-up view" (c_2^1) and some images on which the fish tank is visible (c_3^1). Also for the *banjo* class, we also clearly observe that

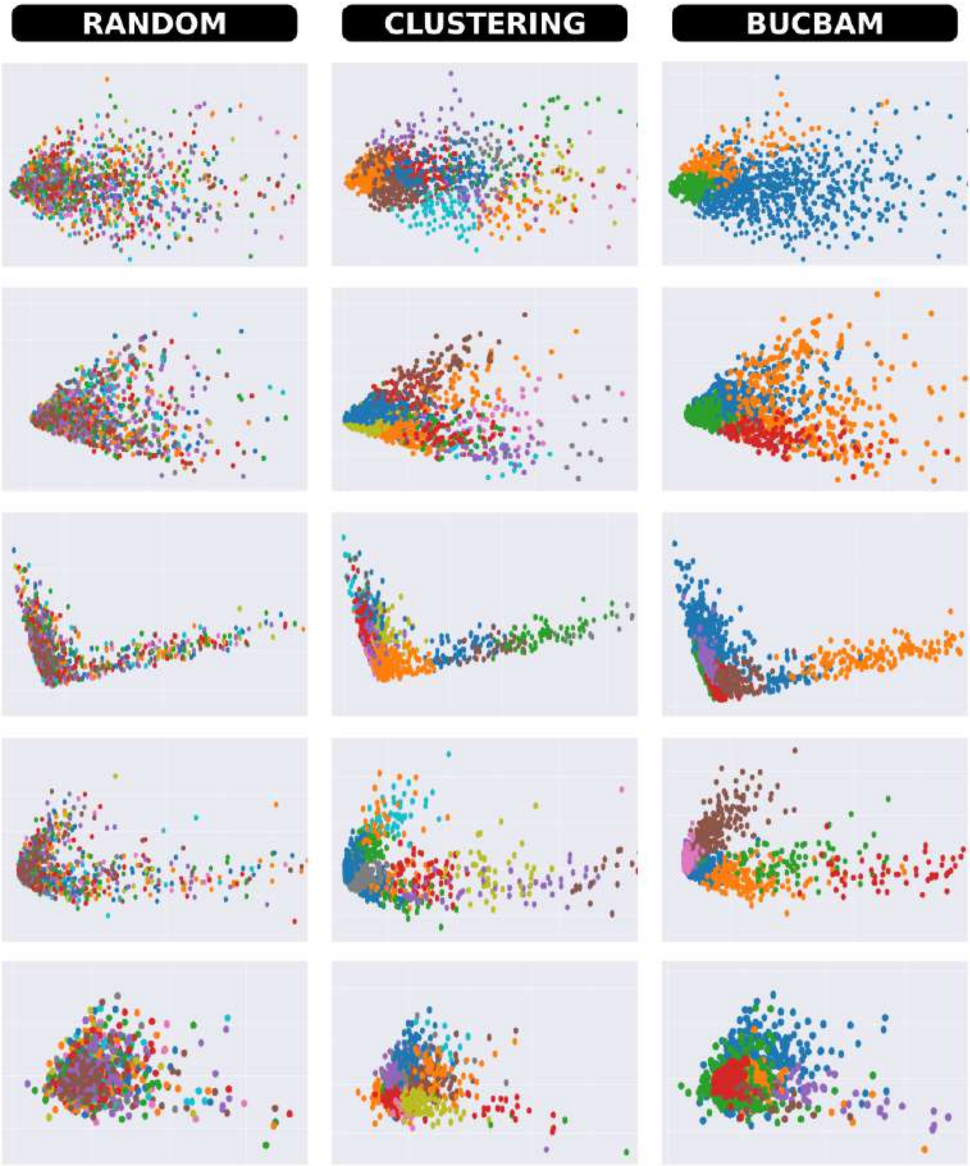

Figure 5: Illustration of the projection of the samples of the clusters on their two first principal components (obtain by PCA) for the three different methods: random split (left) clustering with Kmeans, K=16 (middle) and our BUCBAM proposal (right). In each graph, each color represents a certain cluster. For all the methods, each line represents the instances of a specific category, that is the same than the one represented at the same line in Figure 4. Note that, in contrast to other methods, ours provide clusters with *different* amount of colors for each specific category. Best view in PDF.

our method identified a cluster that represents "person playing banjo" c_1^3 and even "person playing banjo in a concert" c_3^3. Importantly, while the clustering method tend to results in duplicate clusters (*e.g.*, c_2^1 with c_3^1; c_1^3 with c_2^3; c_1^4 with c_2^4 etc.), ours tend to provide only dissimilar results, thank to our merging process.

Finally, we also computed a principal component analysis of the representations of each specific category and projected the vectors on the first two principal components, keeping a different color for each (new) finer category (Figure 5). As expected, with the random split, the vectors are uniformly distributed while the two other methods tend to form some groups. Although these results are qualitative, one can see that the proposed BUCBAM method exhibits slightly more grouped points than the K-means.

References

[1] H. Bilen and A. Vedaldi. Universal representations: The missing link between faces, text, planktons, and cat breeds. *arXiv:1701.07275*, 2017.

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

[3] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *ACM Conference on Image and Video Retrieval*, CIVR, 2009.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge. *IJCV*, 2010.

[6] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *PAMI*, 2006.

[7] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[8] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

[9] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *IEEE Computer Vision, Graphics & Image Processing*, 2008.

[10] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *CVPR*, 2009.

[11] S-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *NIPS*, 2017.

[12] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *CVPR*, 2018.

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[14] Youssef Tamaazousti, Hervé Le Borgne, and Céline Hudelot. Diverse concept-level features for multi-object classification. In *ICMR*, 2016.

[15] Youssef Tamaazousti, Hervé Le Borgne, and Céline Hudelot. Mucale-net: Multi categorical-level networks to generate more discriminating features. In *CVPR*, 2017.

[16] Youssef Tamaazousti, Hervé Le Borgne, Céline Hudelot, Mohamed El Amine Seddik, and Mohamed Tamaazousti. Learning more universal representations for transfer-learning. *arXiv:1712.09708*, 2018.

[17] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset, 2011.

[18] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *ICCV*, 2011.