Supplementary Material for "MuCaLe-Net: Multi Categorical-Level Networks to Generate More Discriminating Features"

Youssef Tamaazousti[1,2] Hervé Le Borgne[1] Céline Hudelot[2]
[1]CEA, LIST, F-91191 Gif-sur-Yvette, France.
[2]CentraleSupélec (University of Paris-Saclay), MICS, 92295 Châtenay-Malabry, France.

Introduction

In this supplementary material, we provide further evidence that supports the quality of the proposed method. These additional experiments were produced using the same version of the approach explained in our main paper and include the following items:

- More details about the implementation of the proposed MuCaLe-Net approach and released material[1] (Sec. A);
- Details about the performances of MuCaLe-Net compared to its component-networks (Sec. B);
- Performances of MuCaLe-Net according to the size of the source training-database (Sec. C);
- Performances of MuCaLe-Net on other target-datasets, especially on the fine-grained categorization task (Sec. D);
- Visualization of more unique filters (for all the layers) generated by each component-network of MuCaLe-Net (Sec. E).

All the figures of this document are best viewed in color.

A. More Implementation Details of the Proposed MuCaLe-Net Approach

In this section, we describe more implementation details of our approach. Especially, we provide more details of: (i) the practical categorical-level re-labeling (Sec. 3.1 of the main paper), (ii) the partitioning protocol (Sec. 3.1 of the main paper), (iii) the initialization process used to get convergence from the VGG-16 network (used in Sec. 5 of the main paper) and (iv) the description of the target-datasets (used in Sec. 5 or the main paper).

[1] http://perso.ecp.fr/~tamaazouy/

Figure 1. Examples of *basic*, *subordinate* and *superordinate*-level words used by Humans to categorize objects in images.

Hierarchical versus categorical-levels

Before digging into the details of the re-labeling and partitioning protocols, we give more details about the definition of categorical-level categories and compare it to those of hierarchical-level categories. In fact, given a hierarchy $\mathcal{H}$ with "is-a" relations ($\mathcal{H} = (\mathcal{V}, E)$ consists of a set $\mathcal{V}$ of nodes and directed edges $E \subseteq \mathcal{V} \times \mathcal{V}$), a hierarchical-level corresponds to the set of all nodes in the same level of the hierarchy. Formally, assuming that none of the hierarchical-level nodes has more than one direct ancestor (*e.g.*, $\forall(v_i) \in \mathcal{V}, Card(\delta_{\mathcal{H}}(v_i)) = 1$)[2] , they correspond to the nodes that have the same amount of total ancestors (*e.g.*, $\forall(v_i, v_j) \in \mathcal{V} \times \mathcal{V}, Card(\{\delta^k_{\mathcal{H}}(v_i)\}_{k=1}^{\infty}) = Card(\{\delta^l_{\mathcal{H}}(v_j)\}_{l=1}^{\infty})$. Thus, the definition is mainly based on the topology of the hierarchy (it contains inconsistent and imbalanced information if the hierarchy is imperfect, which is mostly the case in the real-world). In contrast, a categorical-level is defined by a set of categories from the same type. For instance, the *basic* categorical-level corresponds to the most common words used by Humans to categorize objects. *Subordinate/superordinate* categorical-level corresponds to the words more specific/generic than those of the basic-level. Thus, the definition of categorical-

[2] $\delta_{\mathcal{H}}(\cdot)$ corresponds to the *deductive function* introduced in the main paper, that associates to a category v_i of $\mathcal{V}$ its direct ancestor.

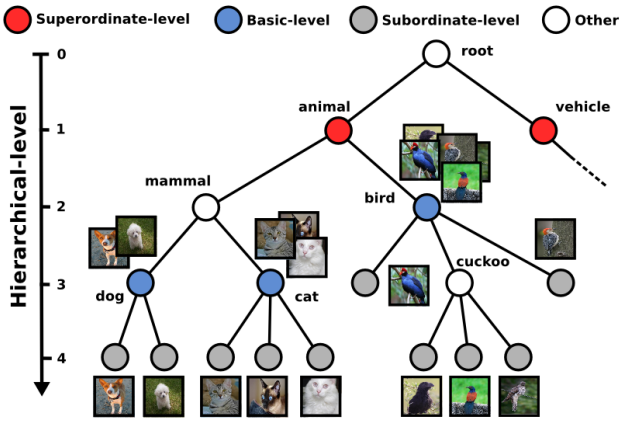

Figure 2. Illustration of the difference between categorical and hierarchical-level categories on a hierarchy with "is-a" relations. Nodes in the same horizontal line belongs to the same hierarchical-level. Colored nodes with the same color belong to the same categorical-level. Blue nodes belong to the *basic-level*, gray nodes to the *subordinate* and red ones to the *superordinate*. By definition, a categorical-level contains the same type of categories (specific only, basic only, generic only) at a given level while a hierarchical one may contain different types at a given level. For instance, the third hierarchical-level contains generic categories (dog, cat, etc.) and specific ones (cuckoo, etc.).

levels is mainly based on a human-cognition knowledge, namely categorization words used by Humans to classify (thus, it contains very relevant and balanced information). In Figure 1, we show some categorical-level words used by Humans to categorize objects and in Figure 2, we illustrate the differences between the definitions of categorical and hierarchical-levels.

Re-labeling protocol

We first describe the near-zero cost re-labeling protocol of the categorical-level categories. As mentioned in the main paper, we used *subordinate*, *basic* and *superordinate*-levels with the initial training-dataset (ILSVRC) labeled at *subordinate*-level (specific such as "rottweiler" or "malinois"). Our goal is thus to re-label the categories of the training dataset in general categorical-levels, namely *basic* (generic such as "dog" or "bird") and *superordinate*-levels (very generic such as "animal" or "vehicle"). A simple way to do that is to associate to each *subordinate* category one of its images (can be any image of the category, as long as it contains only the object and that is clearly visible, which is mostly the case on ImageNet images), show it to the annotator and ask him/her to label the image with the most common word that he/she will use to categorize it generally (*subordinate*) or very generally (*superordinate*). For instance, for the subordinate category *hammerhead*, the annotator will label it by the *basic*-level word *shark* and for the category *weimaraner*, it will label it by the word *dog*. For both *subordinate* categories, it will re-label it with the word *animal* for the re-labeling to *superordinate*-level. Some other re-labeling examples are reported in Table 1. If one wants to have only words of the hierarchy in order to have homogeneity of the re-labeling between the different annotators, he can constrain the annotators by asking them to re-label the images with one of the set of words obtained from the ancestors of the *subordinate*-category.

Subordinate	**Basic**	**Superordinate**
convertible - sport car	car	vehicle
helicopter - fighter plane	aircraft	vehicle
barber chair - rocking chair	chair	furniture
malinois - rottweiler	dog	animal
garfish - puffer - sturgeon	fish	animal
hammerhead - tiger shark	shark	animal

Table 1. Example of re-labeling of subordinate categories (left column) into basic-level ones (middle column) and superordinate ones (right column).

Regarding the re-labeling of the whole ILSVRC dataset ($1,000$ specific categories) that we used in Sec. 5.3 of the main paper, we used an already available list [9] of 483 fine-grained categories labeled to 200 *basic-level* categories and re-labeled (using the above re-labeling protocol) the remaining 517 fine-grained categories. Our re-labeling of the remaining categories results into 280 new *basic-level* categories, with a total of $1,000$ subordinate categories re-labeled in 480 *basic-level* ones. In Sec. 5.2 of the main paper, we have reported some results of our method on ILSVRC$^{0.5}$ with the three categorical-level label-sets, including the *superordinate*-level. To re-label *subordinate* categories into *superordinate* ones, we considered their re-labeled 200 *basic-level* categories (obtained from [9]) as the initial categories and re-label them into *superordinate*. This trick aims us to re-label only 200 categories instead of 483 (*subordinate* categories of ILSVRC$^{0.5}$). This latter, results in 12 *superordinate* categories. The whole sets of *basic-level* categories, *superordinate* ones and their relation to the *subordinate* categories will be made available at http://perso.ecp.fr/~tamaazouy/.

Partitioning protocol

Here, we detail how we automatically get the partitioning of the set $\mathcal{C}$ into G subsets (such that $\mathcal{C} = \bigcup_{i=1}^{G} \mathcal{C}_i$), as described in Sec. 3.1 of the main paper. Once the categories of the generic categorical-level (*basic* or *superordinate*) are given, it is straightforward to partition the set $\mathcal{C}$ into G subsets. In fact, let consider the following set of *subordinate* categories: $\mathcal{C} = \{$*convertible*, *landrover*, *malinois*, *rottweiler*$\}$ and the set of their re-labeling categories $\{$*car*, *car*, *dog*, *dog* $\}$, our method groups specific categories *con-*

Method	airp.	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse
Net-S	84.3	77.4	82.9	80.4	33.6	64.3	84.1	82.1	53.4	57.0	58.8	81.2	82.1
Net-G	82.1	79.2	85.5	77.5	33.2	61.0	84.6	82.6	54.9	52.7	61.5	80.2	82.9
MuCaLe-Net	**84.6**	**80.3**	**86.5**	**80.5**	**36.6**	**65.1**	**85.6**	**84.5**	**56.4**	**59.1**	**63.0**	**83.4**	**84.5**
	mbike	person	plant	sheep	sofa	train	TV	**mAP**					
Net-S	75.8	92.2	48.0	72.9	54.7	75.4	66.2	70.3					
Net-G	74.5	92.0	48.9	69.9	59.1	71.1	66.3	70.0					
MuCaLe-Net	**76.9**	**93.0**	**50.6**	**74.4**	**60.4**	**77.2**	**68.6**	**72.5**					

Table 2. Detailed performance of MuCaLe-Net and its component-networks (Net-S and Net-G) on the 20 categories of the target-dataset Pascal VOC 2007. The networks are based on the AlexNet architecture and trained on the ILSVRC$^{0.5}$ database (half a million images).

Method	airp.	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse
Net-S	91.6	83.3	87.7	84.9	40.1	71.8	87.9	85.5	61.1	**65.4**	73.3	82.3	85.1
Net-G	92.2	82.8	88.7	84.7	41.9	72.8	88.1	85.2	60.8	59.2	71.7	82.9	84.5
MuCaLe-Net	**93.1**	**84.5**	**89.5**	**85.9**	**42.9**	**74.2**	**88.7**	**87.3**	**62.5**	65.3	**74.1**	**84.6**	**86.4**
	mbike	person	plant	sheep	sofa	train	TV	**mAP**					
Net-S	77.6	93.5	51.5	75.9	61.7	90.7	70.8	76.1					
Net-G	78.9	93.5	53.7	70.7	61.8	90.4	70.6	75.8					
MuCaLe-Net	**79.7**	**94.0**	**54.8**	**75.4**	**63.2**	**91.6**	**72.3**	**77.5**					

Table 3. Detailed performance of MuCaLe-Net and its component-networks (Net-S and Net-G) on the 20 categories of the target-dataset Pascal VOC 2007. The networks are based on the AlexNet architecture and trained on the whole ILSVRC database (1.2 million images).

vertible and *landrover* to *car* and *malinois* and *rottweiler* to *dog*, which results in a set of two subsets (thus, $G = 2$) of generic categories with $\mathcal{L} = \{car, dog\}$. It is important to note that, the partitioning is directly based on the set $\mathcal{L}$ of re-labeled categories, thus $G = Card(\mathcal{L})$. From this example, we have a ratio of two specific categories per generic category but in the real-world with real-datasets, the ratio is much higher. Hence, all the images of the specific categories are re-labeled to the generic categories, resulting to *subordinate* categories that may contain much less images than *basic-level* or *superordinate* ones. For instance, as in Figure 2 the *basic-level* category *bird* contains as much images as the amount of images in the ensemble of its subsumed *subordinate* categories.

Initialization process of VGG

Regarding the initialization process used to get convergence from VGG-16 in Sec. 5 of the main paper, we used the weights of a diversified pre-trained network. It is important to note that the initialization process used here is only to solve the technical problem highlighted in the original paper [10] (*i.e*, it is hard to get convergence with any random initialization strategy). In fact, to solve this problem, they used a process hard to re-implement in practice thus, we just used the weights of a pre-trained model. More specifically, we used those of the model pre-trained on the ILSVRC dataset and those of one that we have fine-tuned (from this latter) on a diversified set of $4,000$ categories of ImageNet. Across the experiments, we found that both initialization perform almost equally. For instance, if we initialize MuCaLe-Net with the weights of the model trained on the $1,000$ categories of ILSVRC, we get transferability performance of 91.5 on Caltech-101 dataset and if we use those of the one trained on the diversified categories, we get 92.0. This is slightly higher, however with the same initialization as [10], MuCaLe-Net still improves their performances by 2.7 points of accuracy (91.5 versus 88.8). All component-networks of the proposed MuCaLe-Net (*e.g.*, Net-S and Net-G), both trained with AlexNet and VGG-16 will be made available at `http://perso.ecp.fr/~tamaazouy/`.

Target-Datasets

In this section, we describe with details, the target-datasets used in section 5. of the main paper the main paper. More precisely, we used five popular datasets. Pascal VOC 2007 [2] and VOC 2012 [3] are multi-object datasets, that contain $9,963$ and $22,531$ images respectively, labeled by one or several labels from 20 categories. They are divided into train, val and test subsets. We conduct our experiments on the trainval/test splits ($5,011$/$4,952$ for VOC 2007 and $11,540$/$10,991$ for VOC 2012). Caltech-101 [4] and Caltech-256 [5] are mono-object datasets where images are labeled by one of 102 categories for the former and 257 for the latter. We used popular splits of the literature: 30/60 images per-class for training ($3,060$/$15,420$ in total) and the rest for testing (total: $3,022$/$15,187$) in Caltech-101/256, respectively. Nus-Wide Object [1], is a subset of Nus-Wide and thus a multi-object benchmark. It contains $36,255$ im-

Method	VOC07 mAP	CA101 Accuracy	NWO mAP
Net-S	70.3	79.6	51.2
Net-G	70.0	79.4	51.0
MuCaLe-Net	**72.5**	**82.6**	**54.1**
Net-S	76.1	87.8	62.2
Net-G	75.0	87.2	61.5
MuCaLe-Net	**77.5**	**89.4**	**64.4**

Table 4. Overall performance of MuCaLe-Net and its component-networks (Net-S and Net-G) on three target-datasets (VOC 2007, Caltech-101 and Nus Wide Object). All the networks are based on the AlexNet architecture. On the top of the table we report the results for the networks trained on the ILSVRC$^{0.5}$ database (half a million images) and on the bottom, the results for the networks trained on the whole ILSVRC database (1.2 million images).

ages divided into $21,709$ for training and $15,546$ for testing. Each one is labeled among 31 categories.

B. MuCaLe-Net Compared to its Component-Networks

In this section, we provide more details about the results of the proposed MuCaLe-Net. More specifically, we compare its performances with each of its component-networks, namely Net-S (trained on *subordinate*-level categories) and Net-G (trained on *basic*-level categories).

The comparison is carried out according to a transfer-learning scheme, for which we use three target-datasets (VOC 2007 (VOC07), Caltech-101 (CA101) and Nus-Wide Object (NWO). The overall performances of MuCaLe-Net and its component-networks (trained on ILSVRC$^{0.5}$ or on the whole ILSVRC database) are presented in Table 4. The first salient observation is that the performances of Net-G are below those of Net-S for all the datasets. One could say that, since the target-datasets contains generic categories, Net-G would work better than Net-S, but in practice it is not the case and this is certainly due to the fact that Net-S has been trained with more supervision (the training-images are labeled among a set of low intra-class and high inter-class clusters) than Net-G (the training-images are labeled among high intra-class and low inter-class clusters). As shown in the main paper, the performances of MuCaLe-Net are always above those of its two components, regardless the size of the training-database. This latter, clearly highlights that the improvement does not come from each of the components but from their well-done combination.

In all the experimentation, we always show the overall performances of the methods, without focusing on the per-category performance which can be interesting. Hence, we report the detailed performances (Average Precision on each category) of MuCaLe-Net and its component-networks on the Pascal VOC 2007 dataset on Table 2 for networks trained on the ILSVRC$^{0.5}$ database and Table 3 for networks trained on the whole ILSVRC database. The main observation of this experiment is that, as for the overall performance, MuCaLe-Net always outperforms its component-networks. An exception is for the *cow* category that is slightly more recognizable by Net-S (65.4 versus 65.3). In contrast to the overall performance, Net-G performs slightly better than Net-S on a few categories. This means that high supervision is not always better for all the categories. However, for the whole set of categories, Net-G generally performs lower than Net-S, highlighting the sufficiency to only look at the overall performances to compare the methods.

C. Effect of the Training-Database Size

In this section, we evaluate the effect of the training-database size on the performances of the proposed MuCaLe-Net strategy and compare it to the Standard CNN strategy. The size of a database can be represented by three aspects: (i) the total number of images, (ii) the total number of categories and (iii) the number of images-per-class for each class. In this experiment, we only variate the two first aspects (total number of images and categories) and neglect the third one since it has been shown in [12] that the performances does not increase with roughly more than $1,200$ images-per-class, which is already the amount of images-per-class in the training-databases we use. We thus take the whole ILSVRC database containing around 1.2 million images labeled among $1,000$ categories and extract two subsets: (i) ILSVRC$^{0.5}$ containing around $500,000$ images labeled among 483 categories and (ii) ILSVRC$^{0.6}$ containing around $600,000$ images labeled among 583 categories. From each database, we learn one network following the standard CNN learning-strategy (namely "Standard") and one following the proposed learning-strategy (namely "MuCaLe-Net"). The two methods are evaluated in a transfer-learning scheme on the Pascal VOC 2007 dataset and the results are reported on Table 5.

As expected, the larger training-database we use, the better the methods perform on the target-datasets. More interestingly, the proposed MuCaLe-Net learning-strategy performs always better (for the three database-sizes) than the standard CNN learning-strategy, meaning that our technique is highly robust with regard to the size and the nature of the training-database.

D. Transfer-Learning on Fine-Grained Image Classification Task

In the main paper, we showed that the proposed MuCaLe-Net approach has a good diversification ability, that is useful on a transfer-learning task since it generates a more universal image representation. However, for rea-

#Images	Method	VOC07 mAP	CA101 Acc.
5×10^5	Standard	70.3	79.6
	MuCaLe-Net	**72.5**	**82.6**
6×10^5	Standard	71.4	82.0
	MuCaLe-Net	**73.0**	**84.1**
1.2×10^6	Standard	76.1	87.8
	MuCaLe-Net	**77.5**	**89.4**

Table 5. Overall performance of the standard CNN learning-strategy compared to the proposed MuCaLe-Net on two target-datasets (Pascal VOC 2007 and Caltech-101) trained using the AlexNet architecture with different size of the source training-database. Note that, the three training-databases not only differs by their number of images but also by their nature since they contain images labeled among different set of specific categories.

Method	CUB-200 Accuracy	Stanford-Cars Accuracy
Standard [10]	60.4	37.6
Standard+Flip [8]	61.0	36.5
Standard+Bbox [8]	65.3	n/a
MuCaLe-Net$^{0.5}$ (ours)	71.0	46.4
MuCaLe-Net (ours)	**71.5**	**47.1**

Table 6. Overall performance of MuCaLe-Net compared to the standard learning-strategy in order to show that MuCaLe-Net also generates a more universal image representation for the hard fine-grained categorization target-task. We also compare it to baselines reported in [8], namely "Standard+Flip" (containing flipped images at train and test phases) and "Standard+Bbox" (containing bounding-box annotations at training and testing phases) on two fine-grained classification datasets (CUB-200 and Stanford-Cars). All the networks are based on the VGG-16 architecture and are trained on the whole ILSVRC, except MuCaLe-Net$^{0.5}$ that has been trained on the ILSVRC$^{0.5}$ database.

sons of space and clarity, we only conduct transfer-learning on "classical" image classification benchmarks, that usually contain quite generic classes. In this section we show that the proposed strategy is also useful on datasets containing images labeled among fine-grained categories.

For this, we tested the method on two widely-used fine-grained datasets, namely Caltech-UCSD2011 [11] (denoted as CUB-200) and Stanford-Cars [6]. The former contains 200 bird categories and each has around 30 training images. The latter contains a total of $8,144$ images labeled among 196 car categories, resulting to roughly 40 training images per class. In any case, both datasets contains few training-data per category, which makes sense to deal with it on a transfer-learning scheme.

In this experiment, the goal is to show that the proposed MuCaLe-Net strategy still generates, even on hard fine-grained target-datasets, a more universal image representation (more performing on the target-datasets) compared to the one generated by the standard CNN-learning strategy. For the sake of fair comparison, we evaluate with the same settings as the baselines of [8], that is to say, we use pre-trained models based on the VGG-16 architecture, for the target-dataset images, we extract the penultimate fully-connected layer ($fc7$) of each network of the methods and learn each class with a *one-vs-all* SVM classifier. Note that the standard strategy corresponding to train on specific categories with the pre-trained VGG-16 model [10] has also been implemented by [8], thus we also report the scores they report on their paper for two datasets. A slight difference with our implementation is their augmentation of the training and testing target-data by flipping the images, resulting to a slight improvement of performance. We thus denote their baseline as *Standard+Flip*. They also report for the CUB-200 dataset, the results when bounding-box annotations are provided at train and test phases (we denote it as *Standard+Bbox*), thus we report this result too. The results are presented in Table 6.

From the results, we observe that the proposed MuCaLe-Net is better than all the baselines and more importantly, better than the standard learning-strategy with an improvement of 11.1 points on CUB-200 and 8.8 points on Stanford-Cars. Surprisingly, our method works much better on specific tasks than on generic ones (presented on the

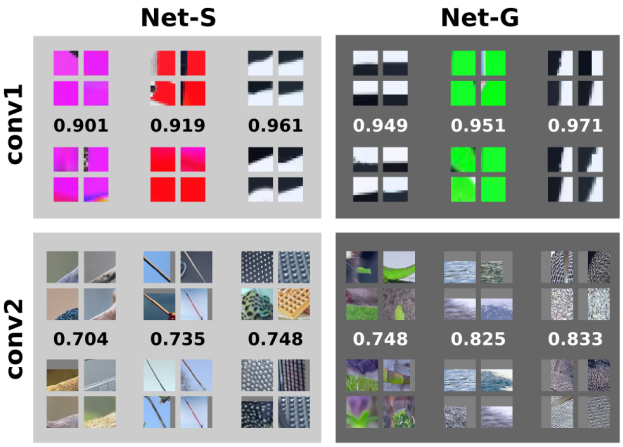

Figure 3. Illustration of pairs of convolutional-filters with a high similarity score. Top part contains the representation of filters (through the top-four image-patches that highly activate them) from $conv1$ and the bottom part, those from $conv2$. In each part, there are two blocks: one for the filters from Net-S (left) and one for those from Net-G (right). For each network (each block), we show its filters on top, the most similar ones from the other network on the bottom and their correlation score. We clearly see that filters that are visually-similar have a high correlation score, meaning that the "correlation" is a good similarity-metric to compare convolutional-filters.

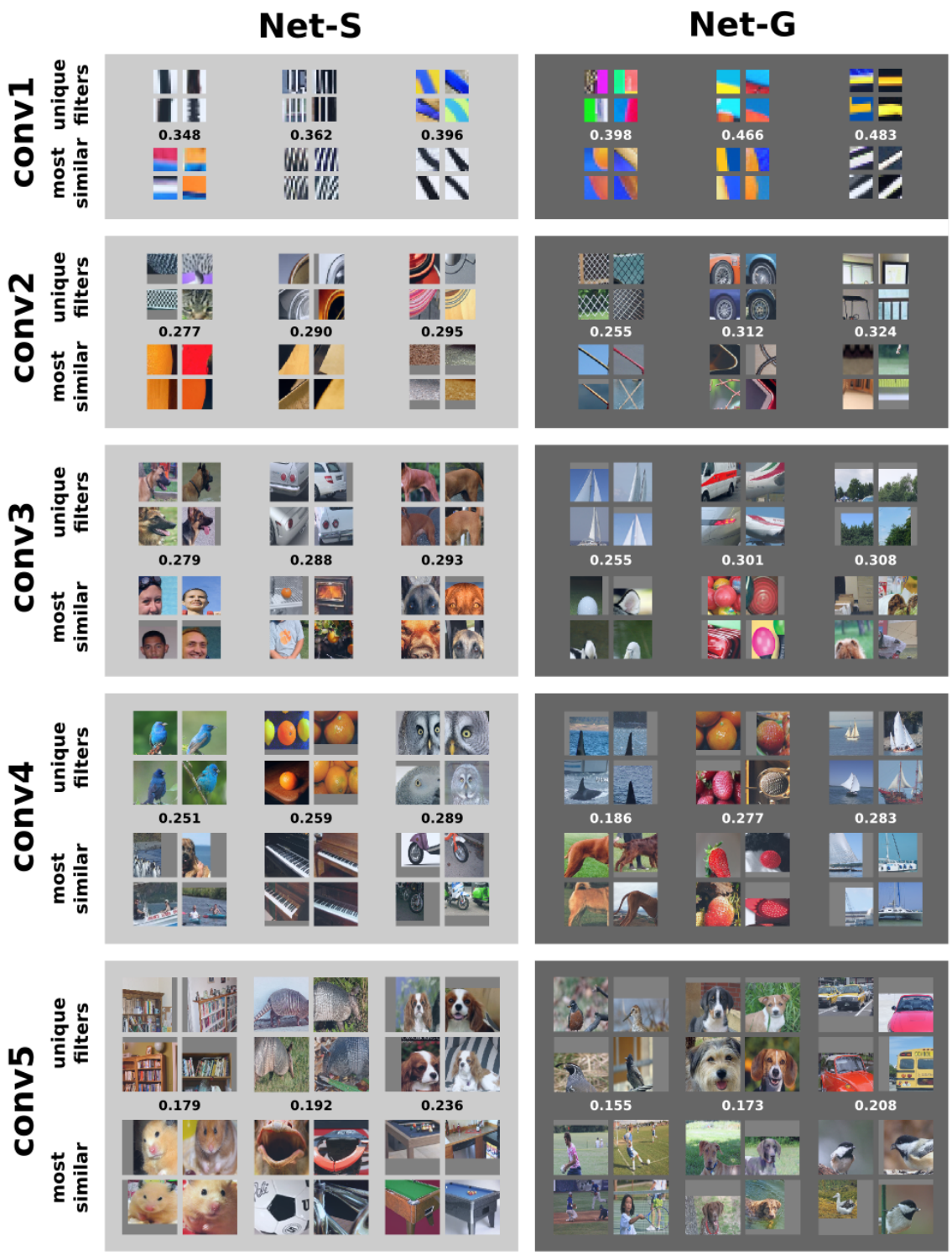

Figure 4. Visualization of unique filters generated by Net-S (left) and Net-G (right) for the five convolutional-layers of the AlexNet architecture. In each of the five blocks we represent on the top part, the filters (through the top-four image-patches that highly activate them) from one network (Net-S on the left and Net-G on the right) on one convolutional-layer and on the bottom part, their closest filters from the other network. Between those filters, we indicate their correlation score. Simply said, in each block (convolutional-layer), the three filters on the top-left and the three on the bottom-right belongs to Net-S and the three filters on the bottom-left and the three on the top-right belongs to Net-G.

main paper), highlighting a high potential on this task. Obviously the performances of the fine-grained methods of the literature are much higher than those we report here. However, here we do not use the well-known techniques to considerably boost the performances, especially fine-tuning the initial models on these target-datasets and using fine-tuned part-detectors to detect and represent the localized parts. In fact, the goal here is only to show that, on a transfer-learning scheme, even on the harder fine-grained target-task, the proposed MuCaLe-Net strategy has a considerable diversification ability (generates more universal image representation) compared to the standard learning-strategy.

E. Visualization of Unique Filters Generated by Component-Networks of MuCaLe-Net

As mentioned in the main paper, we show more visualizations of the unique filters generated by each component-network of the proposed MuCaLe-Net strategy. More precisely, we first show some correlation values between visually-similar filters to highlight the efficiency of the use of the "correlation" metric to compare convolutional-filters, then we show more unique filters generated by Net-S and Net-G on all the convolutional-layers.

In order to assert that the difference highlighted by the two networks is not due to a bad similarity-metric (here we used the "correlation" metric as in [7]), it is crucial to show that when filters are visually similar, the metric outputs a high score (here the maximum absolute value is 1). Thus, we took the two pre-trained categorical-level networks (Net-S and Net-G) and show the patches that highly activate some of the most similar convolutional-filters between the two networks (in term of correlation-metric). These similar filters are presented in Figure 3. We only show the filters from $conv1$ and $conv2$ since, as demonstrated in the submitted paper, above $conv2$-layer the filters are not very similar, thus it does not make sense to show them here. From the results, we clearly observe that when the output value of the similarity-metric is high, the filters are highly visually-similar. In $conv2$, the range of the correlation-scores is slightly below the range of values in $conv1$, but it is still relatively high. In fact, we see that some structure-like (*water*, *dotted*, *green-point*, etc.) filters are very visually-similar as predicted by the correlation-metric. This experiment shows that the correlation-metric is highly suitable to compare convolutional-filters.

Now that we have shown the suitability of the metric we use, we show more visualizations of unique filters generated by Net-S and Net-G. In fact, in Section 4.1 of the main paper, we have shown visualizations of some unique filters from $conv5$, here we show more visualizations and especially for all the convolutional-layers. Figure 4 reports all these new visualizations of unique filters that highlights the five following main points:

- the more we go deeper, the more the filters represent abstract object-parts;
- the two networks (Net-S and Net-G) generate very different filters;
- the more we go deeper, the more correlation-scores of the most similar filters are low. This clearly means that the more we go deeper, the more the filters generated by one network are far from those generated by the other network;
- the filters generated by Net-S are very specific at the deeper layers (conv4 and conv5), for instance it contains very specific breed of dogs, very specific breed of rodent, very specific breed of birds;
- the filters generated by Net-G are very generic at the deeper layers, for instance it contains different breed of dogs, different breed of birds and different kind of fruits.

The first point confirms what is already known [12, 13] and the others confirm what has been highlighted in Sec. 4 of our main paper. To resume, these visualizations show that the similarity-metric we used is well suited to compare convolutional-filters. Above all, it clearly confirms, qualitatively, the diversification ability of the proposed MuCaLe-Net learning-strategy, that is to say, its ability to generate more relevant filters from the same training images.

References

[1] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *CIVR*, 2009. 3

[2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge. *IJCV*, 2010. 3

[3] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2012, 2012. 3

[4] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *PAMI*, 2006. 3

[5] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007. 3

[6] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshop*, 2013. 5

[7] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. Hopcroft. Convergent learning: Do different neural networks learn the same representations? In *ICLR*, 2016. 7

[8] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, 2015. 5

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 2

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 5

[11] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset, 2011. 5

[12] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. In *ICML*, 2015. 4, 7

[13] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 7