An Empirical Study of Feature Dynamics During Fine-tuning

Hamed Behzadi-Khormouji[0000−0003−1546−3852] Lena De Roeck
José Oramas[0000−0002−8607−5067]

University of Antwerp, sqIRL/IDLab, imec
Antwerp, Belgium

Abstract. Fine-tuning is the common practice of adapting a pre-trained model so that the encoded features can be reused for a new target task. Up to now, there have been efforts in the literature to get insights from the fine-tuning process by measuring the internal feature alignment between the source and the target datasets. However, they suffer from three weaknesses. First, their findings are limited to a few datasets, deep models and focused on only one similarity metric which results in discordant observations and doubts regarding reliability. Second, the conducted evaluations are either purely qualitative, which lends itself to subjectivity; or purely quantitative, which suffers from reduced intelligibility. Third, existing analysis focus on the two extremes of the fine-tuning process, i.e. on the model pre and post fine-tuning. In doing so, there is no room for analyzing the dynamics that link these two extremes. Here, we conduct both quantitative and qualitative analyses that aim at shining a light on the feature dynamics during iterative stages of the fine-tuning process. The analysis shows that feature similarity is reduced, even in early stages, between the source model and its fine-tuned counterparts when the target domain is dissimilar. Moreover, it illustrates domain shift across iterations of fine-tuning procedure. We believe the presented methodology could be adopted for the analysis of fine-tuning processes and help pinpoint the reasons why some of these processes are more effective than others. The implementation is available at https://github.com/hamedbehzadi/TransferLearning_Interpretation

Keywords: Model Interpretation · Neural Networks · Fine-tuning.

1 Introduction

Convolutional Neural Networks (CNNs) have shown the valuable ability of discovering patterns and relationships among data. Training these CNNs is still not a trivial task. Despite hardware advances, they require massive amounts of computational power and data to do so. To reduce these resource demands, methods within the umbrella of transfer learning are commonly used. One of the most popular and classical transfer learning methods is fine-tuning [35]. This method was originally introduced as a technique to reuse a pre-trained model so that the already learned features, also known as representation, can be reused for a new

task. This is specially desirable when little labelled training data is available to address a new task. Moreover, it reduces the challenges involved in training a model for a new task from scratch [35].

Despite its advantages and popularity in different fields, little research has been carried on the representation dynamics that occur as part of a fine-tuning process. Efforts related to the study of fine-tuning processes follow one of two directions. On the quantitative side, they include verification of the fine-tuning effectiveness for the image classification through classification performance metrics [14,21,27]. Alternatively, [5,10,21] adopt similarity metrics to measure the distance between internal features of two specific models, a pre-trained model and its fine-tuned counterpart. On the qualitative side, [28] provides visualizations of the internal units during different iterations of the fine-tuning to enable the visual inspection of changes in the internal features.

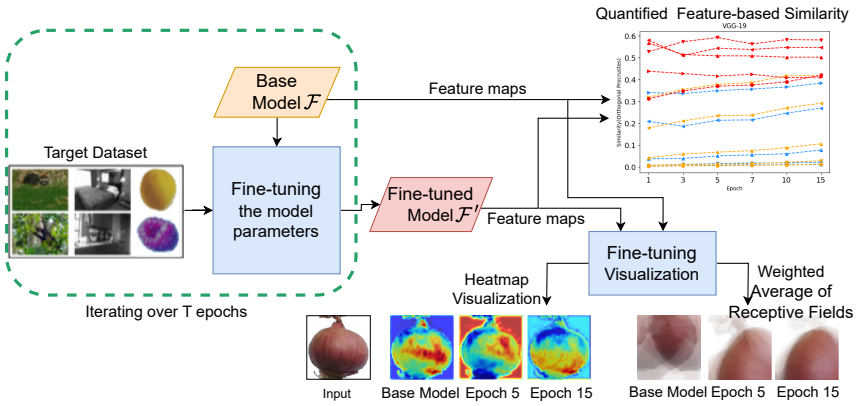

Fig. 1. The pre-trained model is iteratively fine-tuned on the given dataset and task. During this process, at each iteration, multiple feature-based similarity metrics and visualizations are computed from the internal features of the source model and those of the fine-tuned models.

While these works have shed light on the fine-tuning process, they suffer from the following two weaknesses. First, while a plenitude of intuitive notions and hypotheses about the fine-tuning process have been put forward via qualitative analysis, i.e. generating visualizations of the features encoded by the internal units [28], reliance on qualitative visualizations lends itself to subjectivity. Moreover, it does not necessary guarantee generalization in the made observations and neither the proper comparison with respect to other works.

Second, while [10,21] have conducted a quantitative similarity analysis among features, their findings are limited to a few datasets, specific CNN architectures, and are based on only a single feature similarity metric, namely Linear CKA [16]. This not only resulted in discordant observations [16,8], but has not survived the

test-of-time considering the observations made by recent work [7] on the unreliability of linear CKA. Moreover, feature similarity has been measured quantitatively between a pre-trained model and its updated version after a complete fine-tuned process without analyzing the dynamics of features encoded at the different stages of the fine-tuning process. This limits the insights that can be obtained regarding the transferred features. Besides, the evaluation conducted by this group of works focuses on quantitative metrics, thus omitting any underlying qualitative differences that may help getting a better understanding of the fine-tuning process.

Our Proposal. To address the aforementioned weaknesses, this work presents an analysis of intermediate stages of the iterative fine-tuning process for a given model. The followed pipeline includes both quantitative and qualitative analysis (Fig. 1). Compared to efforts in the literature, the conducted experiments include variety of CNN architectures and target datasets. In our quantitative analysis, we consider the Partial-Whitening Shape Metric [33] to quantify feature similarity during fine-tuning process. In the qualitative analysis, different types of visualizations such as activation heatmaps and average visualizations are generated to get an insight on the state of features encoded in the model at a given iteration of the fine-tuning process. These visualizations are produced at the level of individual samples as well as at the class level. The resulting analysis is focused mainly on the inspection of the features encoded in internal units of a given model, more specifically, in the feature extraction part (convolutional layers) of the architecture. In addition, we use these inspection capabilities to study the level to which features are re-used during the different iterations of the fine-tuning process. Hence, the goal of this work is not to push the state-of-the-art further by improving classification performance via fine-tuning, but rather obtaining a better understanding of the dynamics of the fine-tuning process through a variety of data and tasks scenarios. In this regard, the different feature-based similarity measurements and visualizations, included in our work, reveal the influence of data and number of iterations in the fine-tuning process.

This paper is structured as follows: Sec. 2 positions our study w.r.t. existing efforts. This is followed by the description of the followed experimental pipeline (Sec. 3.1 & 3.2) and the different components that are used to quantitatively (Sec. 3.1) and qualitatively (Sec. 3.3) analyze the fine-tuning process. Sec. 4 presents the validation of the pipeline, while closing remarks are given in Sec. 5.

2 Related Work

We describe the existing studies for analyzing the fine-tuning process w.r.t the following four aspects.

Similarity Estimation. In the literature, there is a significant amount of works measuring similarity between layers of different models through different metrics such as Canonical Correlation Analysis (CCA) and its extensions, namely SVCCA [25] and PWCCA [19]. Recently, a newer extension of CCA, called Linear CKA [16] has become a staple metric in the literature [4,22]. Different from

these, here we focus on studying the internal behaviour of a model by comparing the feature similarity between layers of a given model and its fine-tuned counterpart. Along this line, [5,10,21] have utilized Linear CKA for measuring the feature similarity. However, [8] has shown that CCA and its variations are not always statistically reliable for testing functional differences in models. More recently, [6] and [7] investigated the reliability of Linear CKA by measuring the alignment between learned features from the layer before the output of a CNN and transformation versions of them. According to the low similarity values obtained by linear CKA in their experiments, [6] and [7] stress the unreliability of Linear CKA. Moreover, [8] has concluded that the Orthogonal Procrustes metric [30] has not received a comparable attention in the literature, despite its proven track-record. Different from these, our pipeline includes both distance metrics namely Partial-Whitening Shape Metric (PWSM) [33] and Orthogonal Procrustes [30] in a variety of fine-tuning tasks. Additionally, [1] has shown that PWSM, as a Geometry based Shape Metric (GSM), produces reliable results than Linear CKA. Moreover, the mentioned studies examine the similarity of features between a pre-trained model and its fine-tuned version obtained after several iterations of fin-tuning. In contrast, our research delves into the encoded feature similarity between a pre-trained model and its counterparts at different stages of the fine-tuning process, i.e. analyzing the feature dynamics during the fine-tuning process. Finally, different from these works, we verify the coherency among the outputs produced by these two metrics via the Pearson correlation.

Classification Dataset and Architecture coverage. Regarding the CNN architectures considered for the analysis of the fine-tuning process, [10,28] consider Inception-v1 [29] pre-trained on the ImageNet [26] dataset. Also, [21] and [5] analyze the fine-tuning process on Resnet50 [12] pre-trained on ImageNet. Different from these efforts which focus on a single CNN architecture, we consider similar and dissimilar architectures, namely VGG19 [18], Resnet50 [12], and Densenet121 [13] (These two have residual layers and Batch-Normalization layers [12]). All these architectures have been trained on the ImageNet dataset (*Pytorch* library [24]). Regarding the considered classification datasets (i.e., the target dataset), [28] and [5] apply fine-tuning on multiple image recognition datasets. With the exception of these, [10] and [21] have considered only one image recognition dataset. Different from these efforts which focus on the object classification task, we also include the scene classification task via the 15-scene dataset [2]. This is done with the goal of investigating the influence of the task (i.e., object vs. scene classification) in the internal dynamics of the fine-tuning process. Additionally, we use two object classification datasets namely AwA2 [17] (which is similar to the source domain ImageNet, containing similar animal objects and background) and Fruits [20] (dissimilar to the source domain depicting artificial white background). As a result, we consider a large set of data and tasks including 9 models in our experiments (Table 1). We believe this assists obtaining a better picture of the fine-tuning process.

Fine-tuning Visualization. [10] and [28] apply gradient-based methods [23] to maximize the aggregated features of a given unit by computing the feature's

gradient w.r.t the input image, and doing gradient ascent, i.e., Representation Inversion (Feature Visualization) [23]. With the exception of these efforts, existing works only assess the similarity from the quantitative point of view without providing human-understandable visual evidence, i.e. qualitative insights, of the fine-tuning process. On the other hand, while in [10] and [28] the Representation Inversion method resulted in images illustrating synthetic patterns from different classes encoded by the targeted unit, we are interested on identifying real input patterns from the data encoded by the units. Furthermore, we intend to investigate whether the fine-tuning procedure results in a shift in the type of input visual patterns, that is, part(s) of an object, scene, and background, that are exploited for the downstream task of interest. Towards this goal, our pipeline provides a series of visualizations from a given unit, throughout the iterative fine-tuning process, for both individual samples and at the level of each class.

Domain Generalization (DG) & Domain Adaption (DA). DG aims to learn from one or multiple source domains without access to target domains [34]. In other words, a model is trained on one or multiple source training domains to achieve a minimum prediction error on an unseen target domain, while DA is an special case of *Transfer Learning* where, opposite to DG, has access to the target training data [32]. Different from DA in which the source and target domains should be related in such as input and output feature space and tasks [11], fine-tuning can be applied on different source and target domains [35]. Our work focuses on studying transferred internal features during fine-tuning where source and target domains and the tasks of interest might be different.

3 Methodology

The followed pipeline, see Fig. 1, fine-tunes a given source model $\mathcal{F}$ for T epochs. For each epoch $t \in [0, ..., T-1]$, the feature maps of each layer from the source model $\mathcal{F}$ and its fine-tuned counterpart $\mathcal{F}'$ are collected. Afterwards, the similarity between feature maps of corresponding layers from both models is measured. This process is repeated for each epoch and leads to the quantification and visualization of changes of features during the fine-tuning process.

Consider a target image dataset $\mathcal{D}=\{\mathcal{X}^i, \mathcal{Y}^i\}_{i=1}^n$ composed by n image samples $\mathcal{X}^i$ paired with their corresponding class label $\mathcal{Y}^i$. The parameters of the source model $\mathcal{F}$, including layers of the feature extraction and the classifier parts of the source model, are fine-tuned on the target dataset $\mathcal{D}$ which leads to the fine-tuned variant $\mathcal{F}'$. Hence, $\mathcal{F}_l(\mathcal{X}^i)$ and $\mathcal{F}'_l(\mathcal{X}^i)$ compute the feature maps $\mathcal{A}_l^i$ and $\mathcal{B}_l^i$, respectively, from the convolutional layers $l=\{1...L\}$ having a width of w, a height of h, and a depth of d.

3.1 Feature-based Similarity

For each epoch of the fine-tuning process, we collect the feature maps $\mathcal{B}_l^i$. Then, we measure the similarities between each pair of feature maps $(\mathcal{A}_l^i, \mathcal{B}_l^i)$ over n samples for each layer l. This results in a similarity vector $\mathcal{S}_l$ with a shape

$1 \times d$. We consider two metrics: Partial-Whitening Shape Metric (PWSM) [33], determining an angle between the representations (Eq. 1); and Orthogonal Procrustes [30], capturing both magnitude differences and the orthogonality (or alignment) (Eq. 2) between two features.

$$\theta_{PWSM}(\mathcal{A}^i_{l,f}, \mathcal{B}^i_{l,f}) = \arccos\left(\frac{\langle \phi(A^i_{l,f}), \phi(B^i_{l,f})\rangle}{\|\phi(A^i_{l,f})\|_F \|\phi(B^i_{l,f})\|_F}\right) \tag{1}$$

Where ϕ applies a partial whitening function on the feature maps and $||\cdot||_F$ is the Frobenius norm [31]. The suffix f refers to the index of the filter f in the layer l. [33] has shown that this metric is bounded in the interval $[0, \pi]$.

$$\mathcal{S}_{OrthProc}(\mathcal{A}^i_{l,f}, \mathcal{B}^i_{l,f}) = 2 - ||\mathcal{A}^i_{l,f}||^2_F + ||\mathcal{B}^i_{l,f}||^2_F - 2||{\mathcal{A}^i_{l,f}}^T \mathcal{B}^i_{l,f}||_* \tag{2}$$

Where $||\cdot||_*$ is the nuclear [9] norm. [30] has shown that this metric is bounded in the interval $[0, 1]$ for normalized features. For both metrics, a lower value close to zero indicates a lower distance, which means higher similarity. Also, we measure the *Pearson* correlation coefficient between these two metrics.

3.2 Reusability of Pre-trained Features Throughout Fine-tuning

In addition to analyze the representation dynamics during the fine-tuning procedure, we aim to investigate the reusability of pre-trained encoded features from different models during different stages of fine-tuning w.r.t to the task and data domain. This analysis touches an important aspect of fine-tuning through different models that has received close to no attention. To enable the measurement of such reusability, we propose an extension of the procedure described in Sec. 3.1. More specifically, we obtain similarity measurements through Eq. 1 and 2 on the feature maps from different layers for different dataset and tasks. Next, the reusability of features from a given model for the downstream task during fine-tuning can be measured as the mean and standard-deviation of feature-based similarity across different epochs for a given dataset and task.

3.3 Fine-tuning Visualization

In order to investigate the features encoded by the fine-tuned model, after each epoch of the fine-tuning process, two modalities of visualizations are produced for a given convolutional filter f: class-based and sample-based visualizations.

Class-based visualization. Given the feature map $\mathcal{B}^i_{l,f}$ of the sample $\mathcal{X}^i$ from class c computed by the filter f in the convolutional layer l, we compute the receptive field [3] of the filter f at a location (u, v) in the feature map. The location (u, v) is selected by finding the indices of an element with the highest activation (referred as $a^i_{l,f}$) in the feature map. This process results in a region $\mathcal{R}^i_{l,f}$ focused on a part of the input sample $\mathcal{X}^i$. This process is repeated for all samples $\mathcal{X}^i$ from class c. Finally, we compute a weighted average of the obtained regions, i.e., $\sum a^i_{l,f} \mathcal{R}^i_{l,f}$, illustrating the average of real patterns that the filter

f focuses on for the class c. This visualization is generated for each epoch of the fine-tuning process in order to provide qualitative insights on how the encoded features change during fine-tuning. This is used as a complementary evidence to the quantitative results obtained from Sec. 3.1. Figures 5 and 6 illustrate examples of this visualization modality.

Sample-based Visualization. Each feature map $\mathcal{B}^i_{l,f}$, highlighting different aspects of the input, is resized to the size of the input and superimposed on the image $\mathcal{X}^i$ to produce a heatmap visualization of the features. This visualization is generated for each epoch of the fine-tuning process in order to provide iterative visualizations of encoded features per sample. Figures 7 and 8 illustrate examples of this visualization modality.

4 Evaluation

We validate our pipeline on popular CNN architectures; VGG19 [18], Densenet121 [13], and Resnet50 [12], pre-trained on ImageNet [26] (source domain). We consider three classification scenarios where datasets and tasks (target domain) are either similar and dissimilar w.r.t. the domain where the source model was originally trained on. In the first scenario, a scene classification task on the 15-Scene dataset [2] is considered. In the next two scenarios, an object classification task on two datasets AwA2 [17] (similar to the source domain, depicting similar animal objects and background) and Fruits [20] (dissimilar to the source domain, depicting artificial white background) are investigated. We use the Adam optimizer [15] and a batch size of 196, which proved the most optimal in internal tests. Except for the models fine-tuned on Fruits where a learning rate $1e{-}3$ was used, other models were fine-tuned with a learning rate $1e{-}4$ on all the datasets. For reference, Table 1 shows the fine-tuned training and test classification accuracy for the considered datasets and CNNs.

Table 1. Classification performance on train/test sets for different datasets.

CNNs/ Datasets	15-Scene	AwA2	Fruits
VGG19	98.43 / 93.31	96.78 / 89.23	99.83 / 98.42
DenseNet121	99.10 / 94.87	99.57 / 91.59	99.45 / 99.82
Resnet50	98.74 / 93.08	99.43 / 90.97	96.48 / 99.02

4.1 Quantitative Analysis: Internal Feature Similarity

In this section, we conduct a quantitative analysis based on the computed feature-based similarity values (Sec. 3.1), followed by a discussion of the trends observed in different experiments.

Following Sec. 3.1, we use the Partial-Whitening Shape Metric and Orthogonal Procrustes metrics to measure the similarity between internal features (i.e.,

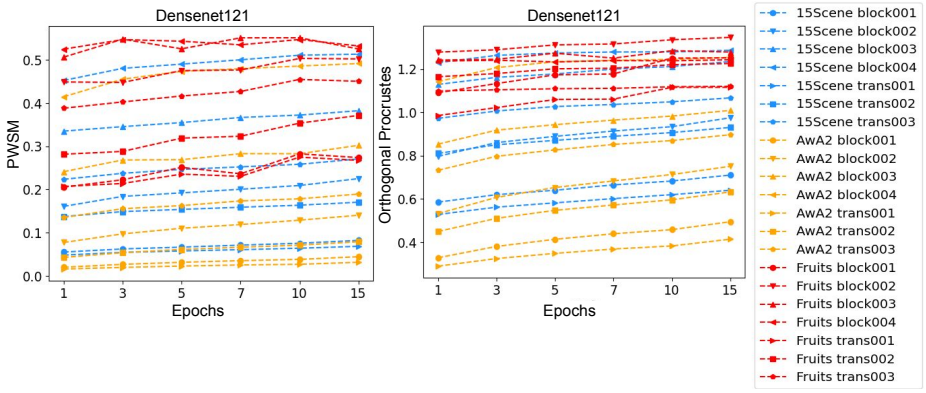

Fig. 2. Feature-based similarity across different fine-tuning epochs between a pre-trained **Densenet121** and its fine-tuned counterparts on the 15-scene (blue), AwA2 (yellow), and Fruits (red) datasets. Similarity is reported based on the **PWSM (left)** and **Orthogonal Procrustes (right)** metrics. Legends show from top to bottom the name of lower to higher layers.

feature maps) from corresponding layers in the source model and its fine-tuned counterparts. In order to be able to monitor the dynamics of the encoded features at different parts of the architecture, we select for our analysis some earlier and latter layers. Specifically, we have considered the $ReLU$ layers located at the end of the major components within the more complex CNNs, e.g. at the end of a dense block in Densenet121. We opted for the focus on $ReLU$ layers given their characteristic, as activation function, of indicating via higher positive values which of the internal features should be propagated to deeper layers and thus contribute to the prediction. Considering this, we discuss observations w.r.t four aspects, (1) the correlation of the similarity metrics, (2) the effect of fine-tuning in the features encoded at different depth levels, (3) feature changes/dynamics across iterations of the fine-tuning process, and (4) the influence of the data involved in the process. These aspects are investigated to assess changes and the re-usability of the features, learned from the source domain, in different fine-tuning scenarios.

Figures 2, 3, and 4 show the similarity across different epochs of the fine-tuning process as measured by PWSM and the Orthogonal Procrustes. Worth mentioning, these metrics measure the distances. As a result, a lower distance value close to zero, indicates a higher similarity. These similarity values are reported between the pre-trained models VGG19, Densnet121, and Resnet50 and their fine-tuned counterparts respectively. In these figures, the curves with similar color refer to the same target dataset. Each point shows the average distance of a given layer, between feature maps from the source model and its fine-tuned counterpart in a given epoch. Lower to higher layers are listed from top to bottom in the legends of the figures.

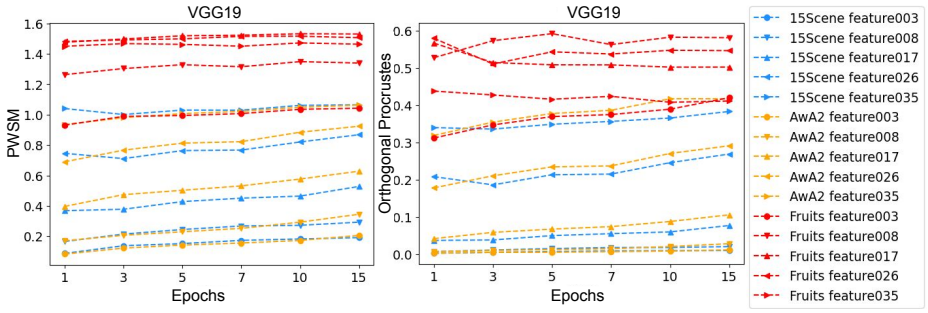

Fig. 3. Feature-based similarity across different fine-tuning epochs between a pre-trained **VGG19** and its fine-tuned counterparts on the 15-scene (blue), AwA2 (yellow), and Fruits (red) datasets. Similarity is reported based on the **PWSM (left)** and **Orthogonal Procrustes (right)** metrics. Legends show from top to bottom the name of lower to higher layers.

Similarity metric correlation. Taking the Figs. 2, 3, and 4 into consideration, both PWSM and Orthogonal Procrustes show extremely similar results over different fine-tuning tasks. Also, we have observed the mean value for Pearson's r as *0.83*, with a standard deviation of *0.43*. This indicates that there is a highly positive linear correlation between these two metrics, in general, they tend to agree with each other.

Effect at different depth levels. We have observed that the fine-tuning process has a different effect depending on the depth of the analyzed features. As can be seen in Figs. 2, 3, and 4, early layers inside a network have consistently lower distance, i.e. higher similarity, to the features encoded on their corresponding layers in the source model (pre-trained on the ImageNet dataset). On the contrary, features encoded at deeper layers have higher distance, i.e. lower similarity to their counterparts in the source models. This indicates that the features encoded in the early layers are slightly modified, when compared to those encoded at the deeper layers. This further confirms the genericity of the features encoded in the early layers and their re-usability during the fine-tuning process. This also hints at a potential solution to improve the fine-tuning process, more specifically, by making the effect of the process proportional to the relative depth of the layers being processed.

Changes across fine-tuning iterations. Considering Fig. 3-right, there is a downward trend in one layer (*feature017*) of VGG19 fine-tuned on the Fruits dataset. Except for this, there can be seen a predominantly upward trend in distance of all of the other layers fine-tuned on different datasets as the models progress through the fine-tuning process (Fig. 2, 3, and 4). This suggests that, while in general the filters converge to a new ideal state for the target dataset,

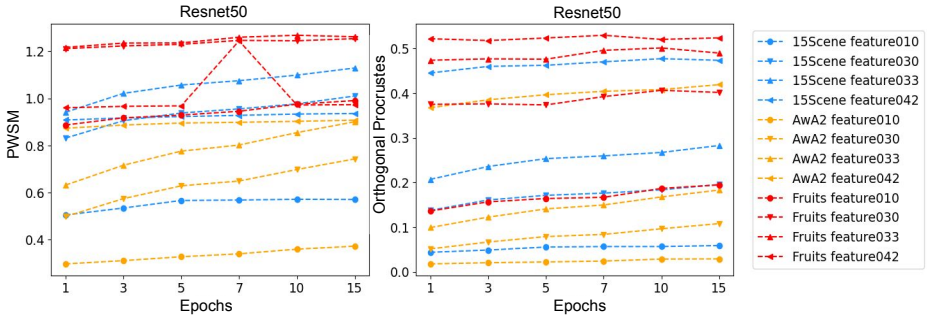

Fig. 4. Feature-based similarity across different fine-tuning epochs between a pre-trained **Resnet50** and its fine-tuned counterparts on the 15-scene (blue), AwA2 (yellow), and Fruits (red) datasets. Similarity is reported based on the **PWSM (left)** and **Orthogonal Procrustes (right)** metrics. Legends show from top to bottom the name of lower to higher layers.

features from the source model require significant changes to adapt to the target domain, hence direct re-usability is somewhat limited.

Data Influence. This experiment investigates the influence that the (dis)similary between the source and target settings may have in the adaptation dynamics of features. With the exception of VGG19 where layers fine-tuned on the 15-scene and AwA2 datasets encode features with closer distance values (Fig. 3), the layers of other models fine-tuned on the AwA2 dataset have lowest distance, i.e. higher similarity, than those fine-tuned on the 15-Scene dataset. Moreover, the similarity of the features learned in these two tasks, i.e., scene classification and object classification, to the features learned in the source domain is higher than that of the object classification task fine-tuned on the Fruits dataset. We can make three interesting observations from these results.

First, from the perspective of the classification task, AwA2 includes classes related to animals which can be found in ImageNet (used for pre-training the models). Therefore, the AwA2 classification task can reuse more features from the source domain, i.e., based on ImageNet, compared to the 15-Scene dataset.

Second, while the 15-scene classification follows a different task, it was able to reuse features learned in the source domain. The evidence for this is the higher similarity trend observed between 15-scene and ImageNet. Besides, images from the 15-scene dataset contain elements in their background that can be found in different classes of the ImageNet dataset.

Third, while the Fruits classification task follows a similar task as that of the source domain (similar to the AwA2 classification task), the images that compose it have a completely different background (i.e., full white background) than those of the source domain (ImageNet dataset). This results in lower similarity between both domains.

To sum up, the source domain has higher influence on the target domain than the tasks used in the pre-training and fine-tuning processes. This also shows that the closest the source and target domains are to each other, the highest amount of features learned from the source domain can be reused for the target domain.

Reusability of pre-learned features during fine-tuning. In this analysis, we summarize the feature-based similarity by computing the average and standard deviation of measured distances across different epochs in accordance with Sec. 3.2 which are presented in Table 2. As can be seen, adapted features from models have the lowest average distances to those from pre-trained counterparts on the AwA2 and 15-Scene datasets with different classification tasks, while they have the highest distances on the Fruits dataset. On the Fruits dataset, which has shown that adapted features have the lowest similarity, these features from Resnet50 are closer to the pre-trained features during iterative stages of fine-tuning compared to those from VGG19 and Densenet121. This reveals that when there is a high dissimilarity between source and target domains, Resnet50 provides a higher amount of features that can be reused.

Table 2. Sensitivity of feature reusability to the task and data domain during fine-tuning procedure.

CNNs / Datasets	AwA2	15-Scene	Fruits
VGG19	**0.345 ± 0.039**	**0.330 ± 0.026**	0.921 ± 0.021
DenseNet121	0.436 ± 0.033	0.576 ± 0.025	0.796 ± 0.023
Resnet50	0.410 ± 0.034	0.550 ± 0.026	**0.749 ± 0.027**

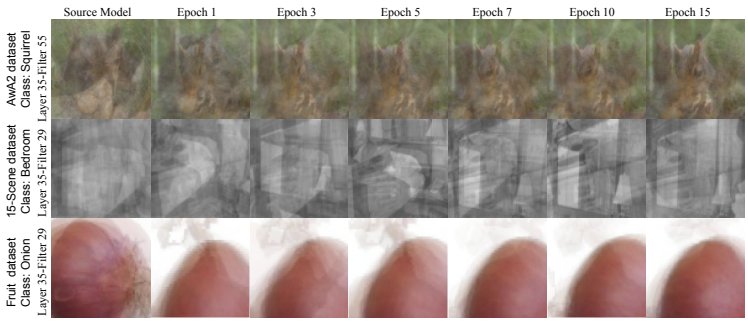

Fig. 5. Average visualizations obtained from **VGG19** (pre-trained on ImageNet) fine-tuned to the AwA2, 15-scene, and Fruit datasets.

4.2 Qualitative Analysis: Fine-tuning Visualization

In this section, we aim to conduct a visual inspection to complement the trends observed in the quantitative analysis (Figs. 2, 3, and 4). Towards this goal, during the fine-tuning process of the considered CNN architectures on the previously-mentioned datasets, we compute sample-based and class-based visualizations (Sec. 3.3) for the features analyzed in Sec. 4.1.

Class-based visualizations. We show examples of the average visualizations produced per class in accordance to the formulations presented in Sec. 3.3. Figures 5 and 6 show the visualizations produced from upper layers in VGG19 and Resnet50 for different datasets, respectively. Each row shows the average visualization, from a specific layer and filter, for a given class. The left-side column shows the visualization from the source model, while the rest of the columns illustrate the visualizations at different epochs of the fine-tuning process.

We can make two observations from Figs. 5 and 6. First, the quantified feature-based similarities from VGG19 and Resnet50 illustrated in Figs. 2 and 4 show, quantitatively, that models fine-tuned on the AwA2 dataset have the highest similarity to the source models. This can be further observed in the first row of the visualization results in Figs. 5 and 6. According to the examples in these figures (Figs. 5 and 6), while the average visualizations from the AwA2 dataset show similar patterns to that of the source model during the fine-tuning process, the average visualizations from the 15-scene and Fruits datasets show patterns differing from those from the source model.

Second, it seems that the relatively lower similarity that were reported on Fruits may find their origin in the fixed white background that characterizes images of this dataset. Specifically in Figs. 5 and 6, the source model, pre-trained on ImageNet, did encode features related to the object of interest of the source domain, i.e. the fruits. However, during the fine-tuning process these features were updated to also include the color transitions introduced by the white background prevalent in the Fruits dataset. This inspection helps to get additional insights on the effect that data may have in the fine-tuning process. It assists on the attribution of the trends observed in the quantitative analysis.

In addition, it can be seen that the patterns emerging in the visualizations change during different epochs. Consider the example of filter seven in layer 54 of Resnet50 fine-tuned on the 15-scene dataset (Fig. 6 middle) for example. In epochs one, three, and five, several persons are shown, while in the rest a single person is depicted. This reveals the effect of the iterative process in fine-tuning as observed in the quantified feature-based similarities in Sec 4.1.

Sample-based visualizations. Opposite to the deeper layers, convolutional layers at lower depth have small receptive fields. This, in turn, makes the inspection of visualizations derived from their receptive fields hard since the small size of the produced visualizations leads to low intelligibility. To address this issue, we use complete feature maps from lower layers, pertaining to each individual sample, as means to highlight important region(s), and the related features,

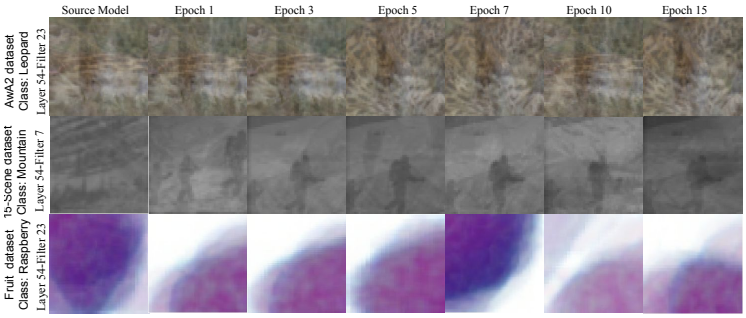

Fig. 6. Average visualizations obtained from **Resnet50** (pre-trained on ImageNet) fine-tuned to the AwA2, 15-scene, and Fruit datasets.

detected by those layers. Hence, here we present some of these sample-based visualizations for the lower layers of a given fine-tuned model based on Sec. 3.3.

Figures 7 and 8 show examples of sample-based visualization for the VGG19 and Resnet50 models fine-tuned on the 15-scene and Fruits datasets, respectively. From these visualizations we can make the following two observations.

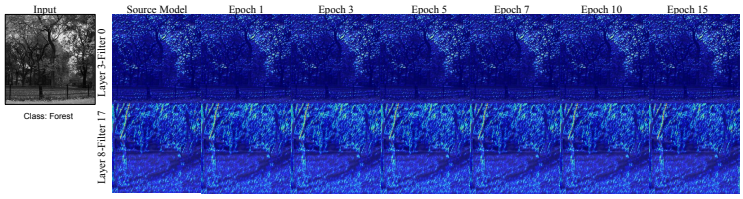

Fig. 7. Examples of sample-based visualizations during fine-tuning from lower layers in a **VGG19** model fine-tuned on the **15-scene** dataset. Each row shows visualizations of feature maps from a filter in a convolutional layer.

First, as can be seen in Fig. 7 for the VGG19 model, filters from the lower layers, i.e., the layers 3 and 8, highlight regions during the fine-tuning procedure which are similar to the regions highlighted by the source model. This further complements what was observed for the quantitative feature-based similarity analysis (Fig. 3) where lower layers have higher similarity to those of the source model. This shows that while the target classification task is different from that of the source model, there is a similarity between low-level features in the source and target datasets which leads to the re-use of some of the low-level features originally encoded in the source model.

Second, opposite to the trend observed in 15-Scene fine-tuning task (Fig. 7), where the considered units are activated in a consistent way during the fine-tuning process; Fig. 8 shows that as fine-tuning progresses, the features shift to new regions of the inputs. This again supports the observation made in the

quantitative analysis, where the target domain has an influence in dynamics of the features during fine-tuning. More specifically in Fig. 8(first row), while the filter 3 from the layer 25 in Resnet50 highlights regions centered on the object, during the fine-tuning process to the Fruit dataset there is a significant shift towards the background features that characterize this dataset. This suggests that this type of transition is not exclusive to features encoded at upper layers, as was observed in Fig. 5 and 6, but also to those encoded at lower layers.

To sum up, considering the highlighted regions in the input features detected by the target model that are similar to those regions detected by the source model along with detected new features during the fine-tuning process reveals the domain shift and feature dynamics across iterations of the fine-tuning process in the lower layers, as initially discussed in the quantitative analysis in Sec. 4.1.

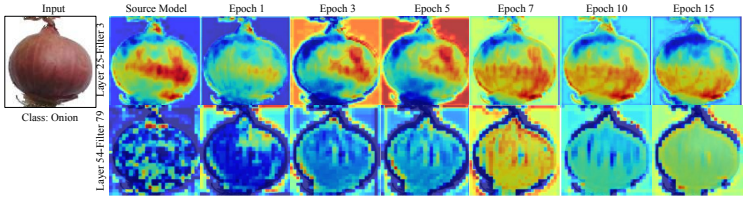

Fig. 8. Sample-based visualizations during fine-tuning from lower layers in a **Resnet50** model fine-tuned on the **Fruit** dataset. Each row shows visualizations of feature maps from a filter in a convolutional layer.

5 Conclusion

This study presents a pipeline to analyse feature similarity between a pre-trained model and its fine-tuned versions from different stages of the fine-tuning process. This pipeline includes feature-based similarity metrics along with activation heatmaps and average of visual patterns. The conducted analysis show a high correlation between feature-based similarity metrics, indicating of consistency among analysis. The analysis reveal that models incorporate more features from the target domain when the source and target domains differ significantly. Moreover, Resnet50 requires the lowest adaption of pre-trained features in presence of dissimilar target data. Moreover, as a future work, we aim to improve the standard fine-tuning algorithm by integrating the obtained insights. One potential approach is instead of fine-tuning the features in a consistent manner across all layers, fine-tuning could be applied on the layers in proportion to their depth location and similarity rate to the source domain.

Acknowledgements: This work is supported by the UAntwerp BOF DOCPRO4-NZ Project (id 41612) "Multimodal Relational Interpretation for Deep Models".

References

1. Acosta, F., Conwell, C., Sanborn, S., Klindt, D.A., Miolane, N.: Evaluation of representational similarity scores across human visual cortex. In: UniReps: the First Workshop on Unifying Representations in Neural Models (2023)
2. Ali, N., Zafar, B.: 15-scene image dataset. Figshare (2018)
3. Araujo, A., Norris, W., Sim, J.: Computing receptive fields of convolutional neural networks. Distill (2019). `https://doi.org/10.23915/distill.00021`, https://distill.pub/2019/computing-receptive-fields
4. Chen, Z., Lu, Y., Hu, J., Yang, W., Xuan, Q., Wang, Z., Yang, Z.: Revisit similarity of neural network representations from graph perspective. arXiv preprint arXiv:2111.11165 (2021)
5. Cui, T., Kumar, Y., Marttinen, P., Kaski, S.: Deconfounded representation similarity for comparison of neural networks. Advances in Neural Information Processing Systems(NeurIPS) (2022)
6. Davari, M., Horoi, S., Natik, A., Lajoie, G., Wolf, G., Belilovsky, E.: On the inadequacy of cka as a measure of similarity in deep learning. In: ICLR 2022 Workshop on Geometrical and Topological Representation Learning (2022)
7. Davari, M., Horoi, S., Natik, A., Lajoie, G., Wolf, G., Belilovsky, E.: Reliability of cka as a similarity measure in deep learning. International Conference on Learning Representations (ICLR) (2023)
8. Ding, F., Denain, J.S., Steinhardt, J.: Grounding representation similarity with statistical testing. International Conference on Neural Information Processing Systems (NeurIPS) (2021)
9. Fan, K.: Maximum properties and inequalities for the eigenvalues of completely continuous operators. Proceedings of the National Academy of Sciences **37**(11), 760–766 (1951)
10. Gonthier, N., Gousseau, Y., Ladjal, S.: An analysis of the transfer learning of convolutional neural networks for artistic images. In: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part III. pp. 546–561. Springer (2021)
11. Guan, H., Liu, M.: Domain adaptation for medical image analysis: a survey. IEEE Transactions on Biomedical Engineering **69**(3), 1173–1185 (2021)
12. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, Proceedings, Part IV 14. pp. 630–645. Springer (2016)
13. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 4700–4708 (2017)
14. Hussain, M., Bird, J.J., Faria, D.R.: A study on cnn transfer learning for image classification. In: Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK. pp. 191–202. Springer (2019)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR) (2015)
16. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International Conference on Machine Learning. pp. 3519–3529. PMLR (2019)
17. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: 2009 IEEE conference on computer vision and pattern recognition (CVPR). pp. 951–958. IEEE (2009)

18. Liu, S., Deng, W.: Very deep convolutional neural network based image classification using small training sample size. In: 2015 3rd IAPR Asian conference on pattern recognition (ACPR). pp. 730–734. IEEE (2015)
19. Morcos, A., Raghu, M., Bengio, S.: Insights on representational similarity in neural networks with canonical correlation. Advances in Neural Information Processing Systems **31** (2018)
20. Mureşan, H., Oltean, M.: Fruit recognition from images using deep learning. arXiv preprint arXiv:1712.00580 (2017)
21. Neyshabur, B., Sedghi, H., Zhang, C.: What is being transferred in transfer learning? Advances in Neural Information Processing Systems (NeurIPS) (2020)
22. Nguyen, T., Raghu, M., Kornblith, S.: Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. International Conference on Learning Representations (ICLR) (2021)
23. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill **2**(11) (2017)
24. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
25. Raghu, M., Gilmer, J., Yosinski, J., Sohl-Dickstein, J.: Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. Advances in neural information processing systems **30** (2017)
26. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**, 211–252 (2015)
27. Shah, H., Park, S.M., Ilyas, A., Madry, A.: Modeldiff: A framework for comparing learning algorithms. In: International Conference on Machine Learning. pp. 30646–30688. PMLR (2023)
28. Szabó, R., Katona, D., Csillag, M., Csiszárik, A., Varga, D.: Visualizing transfer learning. ICML Workshop on Human Interpretability in Machine Learning (2020)
29. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
30. Ten Berge, J.M.: Jc gower and gb dijksterhuis. procrustes problems. new york: Oxford university press (2005)
31. Van Loan, C.F., Golub, G.: Matrix computations (johns hopkins studies in mathematical sciences). Matrix Computations (1996)
32. Wang, M., Deng, W.: Deep visual domain adaptation: A survey. Neurocomputing **312**, 135–153 (2018)
33. Williams, A.H., Kunz, E., Kornblith, S., Linderman, S.: Generalized shape metrics on neural representations. Advances in Neural Information Processing Systems **34**, 4738–4750 (2021)
34. Zhou, K., Liu, Z., Qiao, Y., Xiang, T., Loy, C.C.: Domain generalization: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2022)
35. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q.: A comprehensive survey on transfer learning. Proceedings of the IEEE **109**(1), 43–76 (2020)