Feature discriminativity estimation in CNNs for transfer learning

Victor GIMENEZ-ABALOS [b], Armand VILALTA [a],
Dario GARCIA-GASULLA [a], Jesus LABARTA [b], and Eduard AYGUADÉ [b]
[a] *Barcelona Supercomputing Center (BSC)*
({armand.vilalta, dario.garcia}@bsc.es)
[b] *Universitat Politècnica de Catalunya - BarcelonaTech (UPC)*
(victor.gimenez.abalos@est.fib.upc.edu)

Abstract. The purpose of feature extraction on convolutional neural networks is to reuse deep representations learnt for a pre-trained model to solve a new, potentially unrelated problem. However, raw feature extraction from all layers is unfeasible given the massive size of these networks. Recently, a supervised method using complexity reduction was proposed, resulting in significant improvements in performance for transfer learning tasks. This approach first computes the discriminative power of features, and then discretises them using thresholds computed for the task. In this paper, we analyse the behaviour of these thresholds, with the purpose of finding a methodology for their estimation. After a comprehensive study, we find a very strong correlation between problem size and threshold value, with coefficient of determination above 90%. These results allow us to propose a unified model for threshold estimation, with potential application to transfer learning tasks.

Keywords. transfer learning. machine learning. CNN. feature extraction.

1. Introduction

Convolutional Neural Networks (CNNs) have become the new standard approach for dealing with image processing tasks. These models require exhaustive and expensive training processes, which result in particularly rich and useful representations [1,2,3]. Unfortunately, these methods have strong requirements regarding dataset size, computational power and expert optimisation. For any task in which these factors are an issue, training the model from scratch becomes unfeasible.

Transfer learning studies how to extract and reuse the representations encoded within pre-trained deep neural networks. Among other things, through transfer learning one can exploit deep representations through alternative machine learning methods, without having to train a deep net [4,5,6]. This is also known as *transfer learning for feature extraction.*

Transfer learning for feature extraction is based on performing feed-forward passes through a pre-trained neural network (trained on the **source task**), while feeding it data instances of a new task (the **target task**). In CNNs, this is of-

ten done to transform the data from the image domain into a numerical, high-dimensional space potentially suitable for a wide variety of machine learning methods. This representation, also known as an *embedding*, is typically composed by a subset of internal neural activations. This feature extraction process is independent of dataset size and relatively cheap in computational terms, since it only implies feed-forward passes through the CNN. For the same reason, it requires no hyper-parameter optimisation.

A key factor in the performance of transfer learning solution is the selection of features to extract, and their postprocessing. Given the size of most CNNs, the number of raw features can easily be in the thousands, making it challenging to most machine learning methods due to the curse of dimensionality [7]. To mitigate that, one can reduce the embedding dimensionality by removing features. However this approach implies a vocabulary loss, which is not desirable. An alternative is to reduce the embedding space instead, for example through discretisation [8].

To decide which features to remove or how to discretise the space, it is useful to have a pre-computed measure of feature discriminativity on the target task. This can be done supervisedly, through Kolmogorov-Smirnov distances. In previous work [9] these distances were discretised using two thresholds (one for discriminative feature activation and one for inhibition). Unfortunately, as this method implies recomputing the discriminativity after a random shuffle, it is costly and stochastic.

The goal of this paper is to find an alternative method to discretise the discriminativity space into expected feature behaviour in presence of a class. We base our method on an existing correlation between the average number of instances per class and optimal threshold value. This approach has no stochasticity and reduces the computational cost of the original method. We show how more than 90% of the empirical thresholds' variability is explained by the average instances per class of the target task. Our results indicate that the remaining 10% is strongly correlated with the class imbalance of the target task, and its similarity to the source task.

2. Methods

We focus on the relation between the average number of instances per class in the target task, $\hat{I}_c$, and the optimal feature discriminativity thresholds (t^+ and t^-). $\hat{I}_c$ values for all the tasks considered are shown in Table 1. How we obtain discriminativity thresholds is described next.

To determine feature discriminativity for a target task, we first extract neural activations from the pre-trained model. The first processing done to this raw embedding, is an average spatial pooling on the convolutional filters (to disregard activation location). Afterwards, we perform a feature-wise standardisation across all instances of the target task (*i.e.*, we compute feature-wise z-scores). This results in a standardised embedding.

For each feature f in the standardised embedding, and each class label c in the target task, we define $D_{KS}(f, c)$. This is computed as the signed Kolmogorov-Smirnov (KS) distance between the activations of f on the instances of c (inner-

class), and on the rest (outer-class). $D_{KS}(f, c)$ are values between -1 and 1 indicating the peculiarity of f behaviour for c, either by over-activation or under-activation. We will also refer to this as the **feature-class pair discriminativity**.

The relevant $D_{KS}(f, c)$ values are somewhere between zero (arbitrary feature behaviour) and -1 or 1 (unique behaviour). To maximise the trade-off between noise and information, previous work compared the D_{KS} values obtained for a given class, with those obtained by a set of instances randomly chosen from all target task classes [9]. These values represent the discriminativity value of noise. A second KS test between these two distributions of D_{KS} values provides the threshold that maximises the distance between information and noise. This is done independently for positive and negative D_{KS} values giving rise to t^+ and t^-. The thresholds may be used for classifying the feature-class pairs in the previous work to perform embedding discretisation[8] or for dimensionality reduction.

2.1. Proposed methodology

To avoid the short-comings of the shuffling methodology for obtaining the thresholds, namely the stochasticity and computational complexity, we propose to find a regression model fitting their behaviour based on the target task $\hat{I}_c$. We fit and evaluate such regression using the empirical approach of the shuffling method using several target tasks.

Since we are using the shuffling methodology to guide the regression fitting, we first must make sure that such shuffling results in stable thresholds. For that purpose, we repeat the random shuffling 21 times, resulting in 21 sets of thresholds (see 3.2 for further details). We measure the stability of the shuffling methodology through the standard deviation of these values.

Previous results [9] find high correlation between $\hat{I}_c$ and optimal feature discriminativity thresholds. Analysis of the methodology hints to the presence of a horizontal asymptote as $\hat{I}_c$ increases. At the same time, the absolute value of the thresholds seems to be inversely correlated to $\hat{I}_c$. For this reason, we discard the use of a linear regression. In preliminary studies we considered the following alternatives: the logarithmic, reciprocal and logarithmic reciprocal. The results obtained by the logarithmic reciprocal are remarkably better than the alternatives, which is why these are the only results we show and discuss in the rest of this work. Formally, the logarithmic reciprocal is as follows:

$$t(\hat{I}_c; a, b) = a + b/ln(\hat{I}_c) \tag{1}$$

3. Experiments

Our goal is to find a versatile model for threshold estimation. For this purpose, we study our regression model using two CNN architectures, two source tasks and twenty-one target tasks, since this is the most influential component.

For the CNN architectures we use the VGG16 and VGG19 topologies[10]. These are composed by consecutive blocks of convolution and pooling layers (16

and 19 layers respectively), and two fully connected layers. This sort of architecture is quite representative of the CNN designs being used today. As for the source tasks, we use the following: *ImageNet 2012*[11], a dataset for classification spanning 1000 categories of objects, and *Places 2*[12], a scene recognition task unrelated to *ImageNet 2012* with less categories. Of the possible combinations of architecture-source task, the only case we do not have available is the VGG19 trained on *Places2*. The rest are referenced as follows: VGG16 CNN trained on *ImageNet 2012* (*VGG16IN*), VGG19 CNN trained on *ImageNet 2012* (*VGG19IN*), and VGG16 CNN trained on *Places2* (*VGG16P2*).

3.1. Target tasks

Since we want to obtain a generalisable method, we need to use different target tasks, ideally with different $\hat{I}_c$) and spanning different domains. We consider the following 10 datasets, freely available online:

- *MIT Indoor Scene Recognition* dataset [13] (*mit67*)
- *Caltech-UCSD Birds-200-2011* dataset [14] (*cub200*)
- *Oxford Flower* dataset [15] (*flowers102*)
- *Oxford-IIIT-Pet* dataset [16] (*cats-dogs*)
- *Stanford Dogs* dataset [17] (*stanforddogs*)
- *Caltech 101* dataset [18] (*caltech101*)
- *Caltech 256* dataset [18] (*caltech101*)
- *Food-101* dataset [19] (*food101*)
- *Describable Textures* Dataset [20] (*textures*)
- *Oulu Knots* dataset [21] (*wood*)

To increase the number of target tasks feeding our regression, while providing variance in $\hat{I}_c$, in some cases we consider the different data splits originally provided as different tasks. In particular, we use training sets (TR), test sets (TE), joined training and test sets (TRTE) and validation sets (VAL). From now on, all references to target tasks will regard to a specific dataset and split. The properties of the 21 resulting target tasks are shown in Table 1. Notice the *caltech101TRTE* (followed by *caltech256TRTE*) has a remarkably larger imbalance in the number of instances per class than the rest of tasks.

3.2. Method stochasticity with respect to target dataset

As previously discussed, we need to assess the stability of the shuffling methodology, since we will be using it to validate the consistency of our regression model. Due to computational constraints, we only use a subset of target tasks: *caltech101TRTE*, *mit67TRTE*, *cub200TR*, and *flowers102TR*. This subset spans different topics, have different $\hat{I}_c$, and different imbalance levels. We use the pretrained model *VGG16IN* to obtain the corresponding random $D_{KS}(f, c)$. All details on this experiment are shown in 4.1. These unmistakenly assess the consistency of the shuffling methodology, allowing us to introduce the next experiment.

Table 1. Properties of all tasks used in our experiments, including average number of instances per class ($\hat{I}_c$) and the corresponding standard deviation (Imbalance).

Target task/s	#Images	#Classes	$\hat{I}_c$	Imbalance
caltech101TRTE	9145	102	90	123.07
caltech256TRTE	30607	257	119	85.69
catsdogsTR/TE	3669/3680	37	99/99	1.5/1.5
cub200TR/TE/TRTE	5994/5794/11788	200	30/29/59	0.17/2.91/2.91
flowers102TR/VAL/TE	1020/1020/6149	102	10/10/60	0/0/44
food101TE	25250	101	250	0
mit67TR/TE/TRTE	5360/1340/6700	67	80/20/100	1.39/1.39/0
stanforddogsTR/TE	12000/8580	120	100/72	0/23.12
texturesTR/VAL/TE	1880/1880/1880	47	40/40/40	0/0/0
woodTR	438	7	62	50.84

3.3. Instances per class influence on the threshold

The main hypothesis of this paper is that there is a strong relation between optimal thresholds and target task $\hat{I}_c$, particularly the logarithmic reciprocal function formalised in (1). All the results of the following experiments are shown in 4.2.

Firstly, to study the influence of $\hat{I}_c$ alone, we fit a regression over a single target task: *mit67TRTE*, and pre-trained model *VGG16IN*. This task is balanced in I_c, and we obtain several thresholds by using stratified subsets of the task, corresponding to values I_c multiples of 10 ([10,100]) (Figure 1). To evaluate the goodness-of-fit, we use *leave-one-out cross-validation* R^2 coefficient on these samples. This evaluation method is used with all regressions described in this section.

Secondly, to ensure generalisation of the regression, we fit a regression over the whole set of target tasks on *VGG16IN*, *VGG19IN* and *VGG16P2* separately (Figure 2). In addition, we compare whether these three pre-trained models behave differently, as it may hint to the importance of the model's discriminativity.

Thirdly, we fit a regression on the target tasks that have low class imbalance, using model *VGG16IN* (Figure 3a). We hypothesise that high imbalance behaves differently from the rest, as seen in Section 4.1. Finally, we evaluate this regression with the empirical thresholds of the subsets of *mit67TRTE*, so as to observe if our fitted regression does generalise correctly for newer instances (Figure 3b).

3.4. Practical impact of proposed model

All previous experiments are directed at finding better thresholds for feature discriminativity assessment, and to do so through a trained and reliable model. To assess the impact of such model, we explore its effect on the D_{KS} classified according to these thresholds. Since thresholds determine which feature-class pairs get discretised to either 0, 1 or -1, we perform this evaluation by measuring how many features change value by using the thresholds found by our model. The results are presented in Section 4.3.

Table 2. Threshold statistics for 4 target tasks, ordered by average instances per class descending

Target task	t^- Avg	$t^-\sigma$	t^+ Avg	$t^+\sigma$	$I_c \pm \sigma$
mit67TRTE	-0.109	0.00125	0.119	0.00050	100
caltech101TRTE	-0.140	0.00077	0.160	0.00030	89.66 ± 123.07
cub200TR	-0.174	0.00112	0.195	0.00090	29.97 ± 0.17
flowers102TR	-0.284	0.00234	0.321	0.00238	10

4. Results

4.1. Inner task stochasticity

Table 2 shows the threshold values and their standard deviation for each of the selected tasks. Notice all standard deviations are at least 2 orders of magnitude below the thresholds. This fact speaks for the consistency of the methodology.

4.2. Instances per class influence on the threshold

Figure 1 corresponds to the regression performed on the subsets of *mit67TRTE*. We observe a surprisingly high R^2 coefficient for both positive and negative thresholds. This supports our claim that the thresholds are predictable from I_c.

In Figure 2 we expose the difference in behaviour caused by altering the properties of pre-trained models. In Figure 2a, we compare *VGG16IN* and *VGG16P2* (different source tasks). In Figure 2b we compare *VGG16IN* and *VGG19IN* (different architectures). In both plots of Figure 2 we observe a consistent set of outliers that are not as well adjusted as the others. Remarkably, these correspond to tasks with significant class imbalance (standard deviation above 20, as seen in Table 1). For clarity, these data points have been marked with x in the previous figures. Figure 3a is a regression on *VGG16IN* having removed these tasks: *stanforddogsTE*, *woodTR*, *flowers102TE*, *caltech101TRTE*, *caltech256TRTE*. We refer to this as the *balanced regression*. Figure 3b shows the previously fitted balanced regression, on top of the threshold values from the subsets of *mit67TRTE*.

The R^2 values of all these regressions are presented in Table 3.

4.3. Imbalance error and influence

To evaluate the impact of our methodology, we perform a study on the difference between the original thresholds for *VGG16IN* obtained with the stochastic method, and the predicted with the regression on *VGG16IN* with no filtering. In Table 4 we record the threshold values as well as the percentage of changes. Coherently, the ones with higher amount of changes are the imbalanced tasks, as well as the *food101TE* (this particular case is discussed in Section 5).

5. Discussion

One of our initial hypothesis was that the standard deviation due to the stochasticity is small compared to the values obtained. This seems validated by the re-

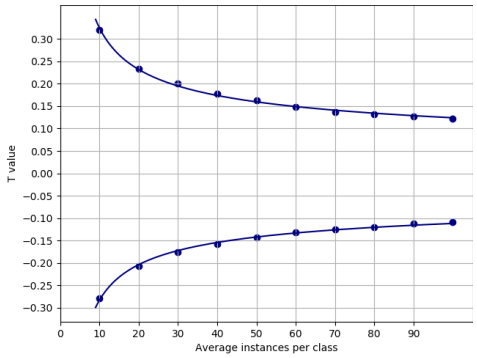
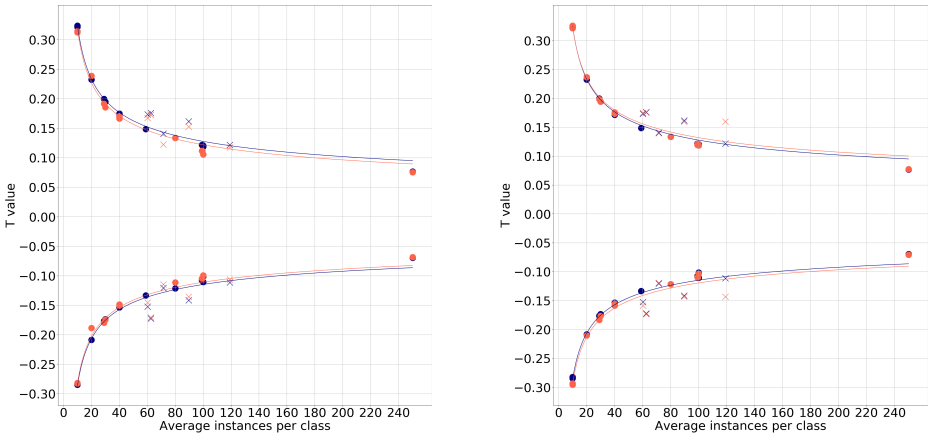


Figure 1. Regression of thresholds for subsets of mit67, with different number of instances per class. The dots correspond to empirical threshold values.

Table 3. R^2 values of each regression. *bal.* stands for balanced regression.

Experiment	t^-	t^+
mit67 subsets	0.986	0.990
VGG16IN	0.944	0.962
VGG19IN	0.920	0.935
VGG16P2	0.936	0.950
VGG16IN *bal.*	0.995	0.997
mit67 *bal.*	0.993	0.995



(a) *VGG16IN* (blue) versus *VGG16P2* (orange).

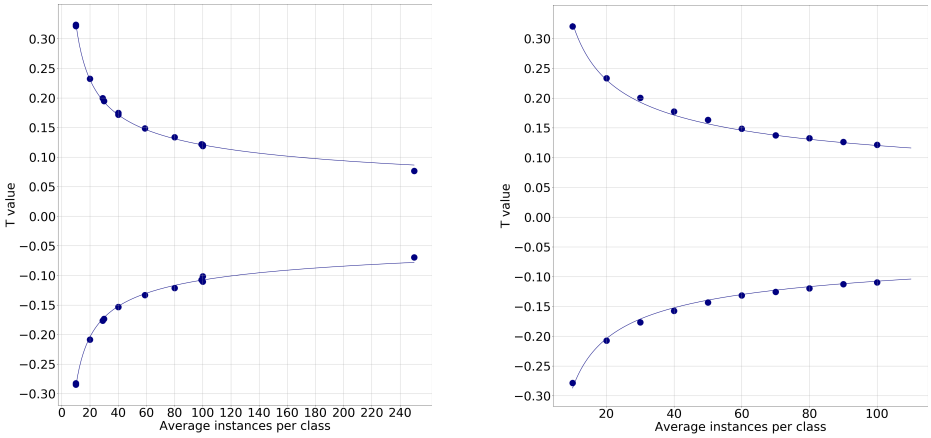
(b) *VGG16IN* (blue) versus *VGG19IN* (orange).

Figure 2. Regression over all target tasks and different pre-trained models. Marked with X are the empirical threshold values of task partitions with σ label distribution above 20.

sults in Table 2 as the standard deviation is two orders of magnitude smaller than the mean threshold value. This table also indicates the presence of an inverse correlation between standard deviation for the thresholds and $\hat{I}_c$ in the balanced tasks. A more complete analysis would be needed to further validate this point.

Regarding the results from Table 3, we observe that the thresholds are highly predictable from I_c. We can attribute part of the error to neglecting class imbalance; notice how by removing the imbalanced tasks we drastically raise the R^2. This extraordinarily high predictability hints to the existence a mathematical relationship. Another significant finding is that the balanced regression (fitted with all balanced tasks) characterises better the *mit67TRTE* subsets' thresholds than the regression tailored for them. We attribute this to the sample size.

Comparing between pre-trained models (Figure 2) we find that regressions are



(a) Regression with dots corresponding to empirical threshold values of the target tasks.

(b) Regression with dots corresponding to *mit67TRTE* cut to different I_c.

Figure 3. Balanced regression.

almost superposed. We hypothesise the difference comes from a different discriminativity across the pre-trained models *w.r.t.* the targets. Even though it seems that a different topology (Figure 2b) yields a greater difference than different source task (Figure 2a), this is actually due to the outlier *caltech256TRTE*. If this task is removed, the difference is much less than that between the source tasks. The impact of both factors (source task and architecture) is thus minimal.

We find an outlier in the balanced regression (Figure 3a and Table 4): *food101TE*. While the task is balanced, the data point is the furthest away from the line, and has the second highest percentage of changes. We think this is caused by this target task being less discriminated against. Unlike other tasks where the average absolute D_{KS} is above 0.2, for this one is 0.15 (near the value of our predicted threshold). This means that feature-class pairs are not very discriminative. To optimise the threshold, the original method lowers the absolute value of the thresholds, raising the amount of noise but also of information. Since there are many feature-class pairs in this interval, small movements of the threshold heavily influence the amount of changes.

6. Conclusions and Future Work

The purpose of this paper was to find a suitable yet simple method to determine thresholds for discriminative noise trade-off in feature extraction. We outline our conclusions next.

1. The stochasticity of shuffling does not heavily modify the thresholds, it only slightly deforms the D'_{KS} distribution (Section 4.1).
2. The number of instances per class can be transformed into the final threshold with the formula 1, obtaining a reduced error (Sections 4.1, 4.3).

Table 4. Threshold influence

Task	t^- original	predicted	t^+ original	predicted	Percentage of group changes
caltech101TRTE	-0.1415	-0.1182	0.1615	0.1317	7.691
caltech256TRTE	-0.1115	-0.1077	0.1215	0.1196	1.016
catsdogsTR	-0.1085	-0.1142	0.1215	0.1271	2.105
catsdogsTE	-0.1075	-0.1143	0.1215	0.1272	2.320
catsdogsTRTE	-0.0825	-0.0915	0.0925	0.1011	3.371
cub200TR	-0.1735	-0.1753	0.1945	0.1972	0.720
cub200TE	-0.1765	-0.1776	0.1995	0.1999	0.276
cub200TRTE	-0.1335	-0.1364	0.1485	0.1526	1.241
flowers102TR	-0.2825	-0.2870	0.3215	0.3254	0.867
flowers102VAL	-0.2845	-0.2870	0.3235	0.3254	0.439
flowers102TE	-0.1525	-0.1353	0.1735	0.1514	5.503
food101TE	-0.0695	-0.0853	0.0765	0.0939	8.668
mit67TR	-0.1215	-0.1228	0.1335	0.1370	0.848
mit67TE	-0.2085	-0.2069	0.2325	0.2335	0.445
mit67TRTE	-0.1105	-0.1140	0.1205	0.1269	1.850
stanforddogsTR	-0.1015	-0.1140	0.1185	0.1269	4.553
stanforddogsTE	-0.1205	-0.1276	0.1405	0.1425	2.098
texturesTR	-0.1535	-0.1570	0.1745	0.1762	0.969
texturesVAL	-0.1535	-0.1570	0.1715	0.1762	1.473
texturesTE	-0.1535	-0.1570	0.1745	0.1762	0.957
woodTR	-0.1725	-0.1336	0.1755	0.1494	10.025

3. Most of the previous error has been shown to come mostly from the class imbalance (Section 4.2).
4. The small remaining error might come from the difference between source and target tasks (*food101TE* in Sections 4.2 4.3).

In this work, we identified potential improvements for future work.

1. The imbalance of the dataset's classes produces an error which we believe could be integrated in the regression function.
2. The difference between VGG topologies seems to be much smaller than between source tasks when considering a pre-trained models. We have yet to see if this applies to the rest of CNN topologies.
3. Information about the real discriminativity, such as the mean absolute discriminativity, might reduce the number of changed features. This would reduce the error in datasets such as *food101*.

Acknowledgements

This work is partially supported by BSC-IBM Deep Learning Center agreement, the Spanish Government through Programa Severo Ochoa (SEV-2015-0493), the Spanish Ministry of Science and Technology through TIN2015-65316-P project and the Generalitat de Catalunya (contract 2017-SGR-1414).

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[3] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, and C. Olah, "Activation atlas," *Distill*, vol. 4, no. 3, p. e15, 2019.

[4] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Factors of transferability for a generic convnet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1790–1802, 2016.

[5] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 806–813, 2014.

[6] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *European conference on computer vision*, pp. 392–407, Springer, 2014.

[7] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE transactions on information theory*, vol. 14, no. 1, pp. 55–63, 1968.

[8] D. Garcia-Gasulla, A. Vilalta, F. Parés, E. Ayguadé, J. Labarta, U. Cortés, and T. Suzumura, "An out-of-the-box full-network embedding for convolutional neural networks," in *2018 IEEE International Conference on Big Knowledge (ICBK)*, pp. 168–175, IEEE, 2018.

[9] D. Garcia-Gasulla, F. Parés, A. Vilalta, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés, and T. Suzumura, "On the behavior of convolutional nets for feature extraction," *Journal of Artificial Intelligence Research*, vol. 61, pp. 563–592, 2018.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[12] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva, "Places: An image database for deep scene understanding," *arXiv preprint arXiv:1610.02055*, 2016.

[13] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 413–420, 2009.

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

[15] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 722–729, 2008.

[16] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012.

[17] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, vol. 2, 2011.

[18] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer vision and Image understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[19] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in *European Conference on Computer Vision*, pp. 446–461, Springer, 2014.

[20] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

[21] O. Silvén, M. Niskanen, and H. Kauppinen, "Wood inspection with non-supervised clustering," *Machine Vision and Applications*, vol. 13, no. 5, pp. 275–285, 2003.