An Analysis of the Interaction Between Transfer Learning Protocols in Deep Neural Networks

Jo Plested(✉) and Tom Gedeon

Research School of Computer Science, Australian National University, Canberra, Australia
jo.plested@anu.edu.au

Abstract. We extend work on the transferability of features in deep neural networks to explore the interaction between training hyperparameters, optimal number of layers to transfer and the size of a target dataset. We show that using the commonly adopted transfer learning protocols results in increased overfitting and significantly decreased accuracy compared to optimal protocols, particularly for very small target datasets. We demonstrate that there is a relationship between fine-tuning hyperparameters used and the optimal number of layers to transfer. Our research shows that if this relationship is not taken into account, the optimal number of layers to transfer to the target dataset will likely be estimated incorrectly. Best practice transfer learning protocols cannot be predicted from existing research that has analysed transfer learning under very specific conditions that are not universally applicable. Extrapolating transfer learning training settings from previous findings can in fact be counterintuitive, particularly in the case of smaller datasets. We present optimal transfer learning protocols for various target dataset sizes from very small to large when source and target datasets and tasks are similar. Our results show that using these settings results in a large increase in accuracy when compared to commonly used transfer learning protocols. These results are most significant with very small target datasets. We observed an increase in accuracy of 47.8% on our smallest dataset which comprised of only 10 training examples per class. These findings are important as they are likely to improve outcomes from past, current and future research in transfer learning. We expect that researchers will want to re-examine their experiments to incorporate our findings and to check the robustness of their existing results.

Keywords: Transfer learning · Convolutional neural networks

1 Introduction

Transfer learning in neural networks generally involves pre-training weights on a large source dataset then applying these weights to a problem on a related target dataset. In this paradigm either:

T. Gedeon et al. (Eds.): ICONIP 2019, LNCS 11953, pp. 312–323, 2019.
https://doi.org/10.1007/978-3-030-36708-4_26

- all but the last layer of weights are frozen, essentially using these weights as a feature detector for another classification algorithm [2,4,15], or
- fine-tuning is performed where all or some of the weights are retrained to fit the target dataset and the original pre-trained weights act as a regularizer to prevent overfitting [1,5,8,9,12,17].

The former is generally used when the target dataset is small and the latter when it is larger.

Deep convolutional neural networks (CNNs) have revolutionized computer vision [10]. Since then a large body of research has shown that performing transfer learning by pre-training a CNN on a large dataset like ILSVRC2012 [3], and fine-tuning the pre-trained weights on a related target dataset, can improve results on a wide range of target tasks [5,7–9,11,12,16,17].

The most systematic and thorough analysis of transfer learning on CNNs to date is the work of Yosinski et al. [19]. They showed that transferring weights pretrained on a related dataset and then fine-tuning them, results in networks that generalize better than those trained directly on a large target dataset. They also showed that when fine tuning is done with standard hyperparameters, routinely used for training from scratch (initial learning rate of 0.01, decay 0.1 every 30 epochs), accuracy on the target dataset increases as the number of layers transferred also increases.

The results of Yosinski et al. have been regularly adopted as an accepted best practice paradigm for transfer learning with CNNs, notwithstanding differences from Yosinski et al.'s target dataset size. There are many examples in the literature that have adopted this practice of all but the final classification layer being transferred with training settings identical or similar to those outlined above [6,9,12,14,17,18]. These settings have been used to:

- show when transfer learning is effective [6],
- compare transfer learning with other methods for small target datasets [14], and
- question whether transfer learning is useful at all [6,14].

Other works ostensibly appear to back up the conclusions adopted by the community from Yosinski et al.'s results. However, the results were only demonstrated to apply under very specific experiment conditions. For example Azizpour et al. [2] used transferred layers of weights as a feature detector with a linear Support Vector Machine (SVM). They showed that in similar tasks all except the final layer of a CNN should be transferred and for even the least related image tasks all but the final two or three layers should be transferred. Their protocol differs from current standard transfer learning practices in that they performed no fine tuning on the transferred weights, and they did not replace and reinitialize the layers not transferred. Replacing more layers with an SVM results in fewer parameters. This likely biases the experiments towards reporting higher accuracy with more layers transferred, particularly with larger target datasets.

Recent work drives this transfer learning paradigm further by pre-training on datasets that are 6× [7], 300× [16], and even 3000× [11] larger than ILSVRC2012. While this body of work demonstrates significant improvements on image classification transfer learning tasks with large target datasets the opposite has often been shown for less related tasks or smaller target datasets [6,14]. This led us to question whether transfer learning protocols developed in [19] and widely adopted by the community are actually best practice for target datasets that are smaller than those used for the experiments performed by Yosinski et al. To answer the above question this paper explores the relationship between:

- the optimal number of pre-trained layers to transfer to a target task,
- fine-tuning hyperparameteres including initial learning rate for each, and layer and learning rate decay settings
- the size of the target dataset.

We show that the interaction between these settings results in the optimal number of layers to transfer being overestimated or underestimated, depending on the size of the target dataset, if transferred weights are trained for too many epochs at too high a learning rate on the target dataset. This relationship is accentuated for smaller datasets and can result in transfer learning showing no improvement or even a negative effect when performed with the protocols adopted from [19] as was shown in [14].

In this paper we make several contributions. We show:

1. There is an interaction between optimal number of layers to transfer and fine-tuning hyperparameters.
2. Freezing layers of pre-trained weights after transferring them rarely produces the best results.
3. Using optimal transfer learning protocols we developed, based on the outcomes of our experiments, produces large changes in results compared to the commonly adopted existing protocols. This particularly applies when working with smaller target datasets. This is likely to impact results that have compared transfer learning, using commonly adopted protocols, with other methods.
4. Optimal transfer learning protocols cannot be predicted from existing research.

We also provide optimal transfer learning protocols for use with a range of target dataset sizes from very small to large where the source and target datasets are similar.

2 Experiments

Our goal is to extend existing work on transfer learning in particular the seminal work of [19] and to identify limitations of its application. We aim to develop transfer learning and fine-tuning protocols based on the size of the target dataset. Given our intention is to compare transfer learning paradigms only we have

used a standard Pytorch [13] implementation of the model Alexnet [10] with no modern architectural improvements. This model is widely used in existing works, allowing our results to be more universally comparable and applicable.

We followed the same experiment protocols as [19]. We split the 1000 object classes in ILSVRC2012 randomly into 500 classes for pre-training with the remaining 500 classes acting as the target dataset. We repeated this process four times for each experiment to create four separate splits of 500 classes for pre-training and 500 classes for the target dataset. We refer to the target dataset that includes all training examples available for each of the 500 random classes as full. The full target datasets have around 645,000 training examples and a minimum of 1,000 per class. We then created different sized target datasets by drawing the first x number of training examples per class to create datasets of size 500x. We selected the first x per class to ensure the same training examples were used across different target dataset sizes and class splits. The other target dataset training set sizes have 500, 250, 100, 50, 25 and 10 training examples per class. We use all the standard ILSVRC2012 validation examples from each random set of 500 target dataset classes for testing. This means there are 50 test examples per class.

3 Results and Discussion

We performed six sets of experiments. The first experiments varied the number of layers transferred for each of a standard set of target dataset sizes and is discussed in Sect. 3.1. Experiments where we varied the number of layers trained at an initial high learning rate are presented in Sect. 3.2. Section 3.3 revisits the experiments varying the number of layers transferred incorporating the best settings from the experiments in Sect. 3.2. Results from experiments varying the number of epochs before the initial high learning rate is decayed are discussed in Sect. 3.4. Experiments to determine the optimal number of layers to freeze are presented in Sect. 3.5. Finally the results using the optimal transfer learning protocols derived from all previous experiments are compared to results using commonly adopted protocols from [19] as well as training from scratch with no transfer learning in Sect. 3.6.

3.1 Vary Number of Layers Transferred for Different Sized Target Datasets

Figure 1 shows the results of all our transfer learning experiments varying the number of layers transferred for our standard set of target dataset sizes while keeping the training hyperparameters constant. The training hyperparameters used were as per [19] with an initial learning rate of 0.01 for all layers reducing by 0.1 every 30 epochs. These results show that for the full target dataset size the accuracy increases slightly with the number of layers transferred. This is in keeping with the outcomes of [19]. However, as soon as the size of the dataset is halved this result no longer holds. It is better to transfer only five layers rather

Fig. 1. Number of layers transferred vs dataset size. Each bar represents the average accuracy over the validation set for a model fine-tuned on one size of target training dataset with a particular number of layers of pre-trained weights transferred.

Table 1. Overfitting on small target datasets

Layers	Training examples per class				Average training and validation loss difference
	Loss	10	25	50	
4	Training	1.55	1.22	1.35	2.91
	Validation	5.07	4.27	3.51	
5	Training	0.71	0.82	1.06	3.63
	Validation	5.38	4.43	3.67	
6	Training	0.49	0.60	0.83	3.90
	Validation	5.34	4.48	3.81	
7	Training	0.39	0.48	0.71	4.72
	Validation	6.26	5.18	4.29	

than all seven as per standard practice. This result is most pronounced for the smallest dataset where accuracy increases from 20.9% when transferring all seven layers to 23.4% when only transferring five layers. This is an increase of 12.2% compared to the original accuracy. These results show that for smaller datasets:

- transferring the correct number of layers has an increasing impact on accuracy, and
- transfer learning in general increases in importance as training from scratch is no longer viable.

Another surprising result is that for smaller target datasets overfitting is much more pronounced when more layers are transferred to the target task. This is shown in Table 1. The aim of pre-training is to act as a regularizer and reduce overfitting. Contrary to this, on a small dataset transferring more layers of pre-trained weights exacerbates the overfitting problem compared to transferring less layers. This finding is more intuitive than it sounds. With smaller target datasets the model does not see enough training examples in the fine-tuning stages to make significant changes to the large pre-trained weights. This leaves

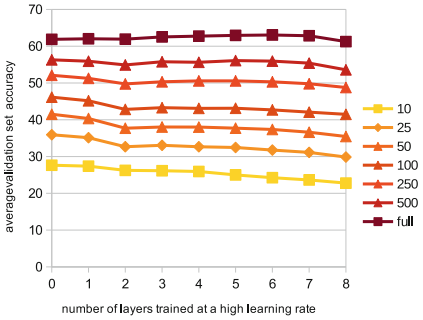

Fig. 2. Number of layers trained at a high learning rate. The degradation in performance, across all target dataset sizes, as more layers are trained with a high initial learning rate is shown.

it with features that are not well suited to the target task, so when aggressive training hyperparameters are used the model overfits to the small training set.

3.2 Vary Number of Layers Trained at a High Learning Rate Initially

Starting from the final classification layer, this experiment varied the number of layers trained at an initial high learning rate of 0.01 for 30 epochs before decaying it by 0.1. The remaining layers were trained with the same regime but using an initial learning rate of 0.001. The number of layers transferred was set at 6 as it produced either the top or comparable to the top accuracy across all target dataset sizes in our other experiments. Figure 2 shows the results of this experiment across the full set of target dataset sizes. It demonstrates that accuracy increases as the number of layers trained at a high learning rate decreases, for all target dataset sizes. This indicates that the fine-tuning hyperparameters used in [19] and in much of the literature since, are too aggressive for the cases examined in our work where source and target datasets and tasks are similar. Another point to note about the results shown in Fig. 2 is that there is a noticeable drop in accuracy when the final two layers are trained with a high learning rate compared to either one or three layers. This is interesting, as it coincides with the number of layers reinitialized. This again highlights overfitting due to aggressively training with features that are not well suited to the target task.

3.3 Compare Number of Layers to Transfer with Low and High Initial Decay epochs

We replicated experiment 3.1 examining the number of layers to transfer, this time only the final layer was trained at a high learning rate initially. We performed the experiment with a high setting of 30 epochs before the initial high learning rate was decayed. We then repeated this with a low setting of five epochs

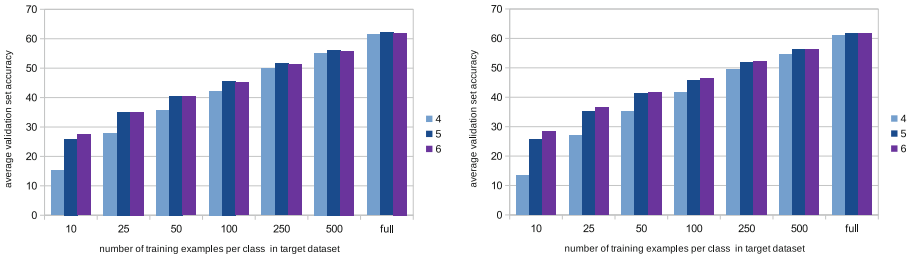

Fig. 3. Number of layers to transfer with low and high initial decay epochs. Comparison of the change in the average accuracies across number of layers transferred with a low (5) left and high (30) right initial number of epochs before decay. All experiments are performed with only the final classification layer trained at a high learning rate initially.

The chart on the left in Fig. 3 shows that with a high setting for initial decay epochs the optimal number of layers to transfer is five, for all but the two smallest target datasets. However, when a low setting for the initial decay epochs is used the optimal number of layers to transfer is six, for all target dataset sizes except the largest. The lower initial decay setting also results in higher accuracies for all but the largest dataset size. The biggest increase in accuracy is 3.5% on the smallest dataset. Whereas the increase on the second largest dataset is only 0.07% and the accuracy on the largest dataset is lower when a small number of epochs before decay are used.

The findings from this experiment are significant as they show the interaction between number of layers to transfer and fine-tuning hyperparameters when pinpointing the optimal transfer learning protocols. This result can also be observed by comparing Figs. 1, 2 and 3. When all layers are trained with an initial high learning rate of 0.01 for 30 epochs, transferring four layers rather than five or six produces only marginally lower results. However, when just the final classification layer is trained with a high learning rate initially, transferring only four layers produces significantly lower results. Both these observations show that if initial learning rates and decay epochs are set too high the optimal number of layers to transfer will be overestimated or underestimated depending on the size of the target dataset.

3.4 Vary Initial Decay Epochs

Our hypothesis was that training for 30 epochs initially at a high learning rate of 0.01 is too aggressive for fine tuning. To test this, we varied the initial decay epochs between 0 and 30 while fixing the number of layers transferred at six. The results of these experiments are shown in Fig. 4. Confirming this hypothesis our results show that training for 30 epochs initially at a high learning rate of 0.01 is too aggressive when fine-tuning on a target dataset and task that is very similar to the pre-training dataset. This continues the pattern noted in the previous experiments that an aggressive training scheme causes overfitting.

Fig. 4. Initial decay epochs. Changes in the average validation set accuracy are plotted, across all target dataset sizes, as the number of epochs before decay are varied for the high learning rate on the final classification layer.

This experiment shows that the pattern is evident even when only training the final classification layer heavily. A surprising result in these experiments is that as the size of the target dataset decreases the number of epochs to train the classification layer at a high learning rate increases. An unexpected curve can be seen in the optimal number of epochs to train at a high learning rate before decay for each target dataset size. These epochs are as follows:

- for the smallest dataset 10 epochs,
- for medium sized datasets with 50, 100 and 250 training examples per class 0 to 3 epochs, and
- for the largest dataset 30 epochs is optimal.

This may seem counterintuitive, however, the very small number of training examples per epoch for the smallest datasets needs to be considered. When this is taken into account these results once again show the effects of overfitting when fine-tuning transferred weights on smaller target datasets. Our smallest target dataset of 10 training examples per class has 5, 10 and 25 fewer batches per epoch than our target datasets with 50, 100 and 250 training examples respectively. Because of this the weights are changed far fewer times during each training epoch. This means that training for 10 epochs on our smallest dataset results in the same number of weight updates as training for only 1 epoch on a dataset with 100 training examples. This further extends our findings that the fine-tuning hyperparameters commonly adopted from [19] are too aggressive to produce the optimal results, particularly for smaller datasets. The weights for smaller datasets should be updated less times at a high learning rate to achieve the best accuracy.

The optimal number of epochs, to train the final classification layer at a high learning rate, increases significantly when fine-tuning with the full set of target dataset training examples. This is in keeping with the discussion above. With a large dataset there are enough training examples to ensure that:

Fig. 5. Number of frozen pre-trained layers. The effect on performance of the number of pre-trained layers of weights frozen during fine-tuning are shown across various target dataset sizes.

- fine tuning the transferred weights at lower learning rates changes the features used for classification enough, and
- performing classification with features that are not optimal for the task does not result in as much overfitting as for smaller datasets.

For this reason, it is not possible to use optimal settings for transfer learning with very large target datasets to perform transfer learning with smaller target datasets or predict the latter from the former.

3.5 Vary Number of Frozen Layers

Current practice for transfer learning with CNNs generally involves locking or freezing the weights of some of the lower layers of a pre-trained CNN while fine-tuning on the target task to prevent overfitting. We tested the assumption that this practice improves performance in transfer learning on CNNs, particularly for small datasets, by varying the number of layers frozen during fine-tuning while keeping the number of layers transferred and hyperparameters constant. The results of the experiments in Fig. 5 show that freezing a high number of layers results in a significant increase in accuracy for very small target datasets of 10 and 25 training examples per class. However, once the target dataset increases to 50 training examples per class, freezing layers has very little effect and freezing more than one layer quickly becomes detrimental as the size of the target dataset continues to grow. When considering these results for freezing layers it should be kept in mind that the source and target datasets for these experiments are as similar as possible, both being drawn from ILSVRC2012, and the tasks, image classification, are identical. We expect that freezing layers will be less useful for very small target datasets and more detrimental for anything larger, for less related source and target datasets and/or tasks. We conclude that freezing layers has very limited scope to improve results on transfer learning.

3.6 Optimal Results

Figure 6 shows the results of using optimal number of layers to transfer and hyperparameters for fine-tuning when performing transfer learning. These results

show that while using the optimal transfer learning protocols improves results for all dataset sizes it becomes particularly important as the size of the dataset decreases. For smaller datasets:

- transferring the optimal number of layers and using the best hyperparameters has an increasing impact on accuracy, and
- the importance of transfer learning increases because training from scratch is no longer viable.

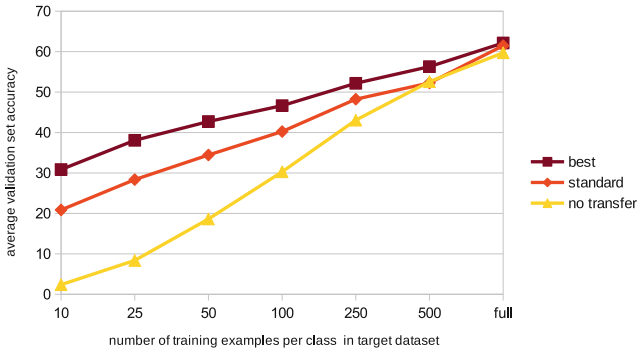

Fig. 6. Optimal transfer learning protocol vs commonly used vs no transfer learning. The final results of using the optimal transfer learning protocols we developed compared to those commonly used in previous research and no transfer learning

Table 2. Optimal transfer learning settings

	Training examples per class						
	10	25	50	100	250	500	Full
Layers to transfer	6	6	6	6	6	6	5
Initial decay epochs	10	5	3	3	3	0	30
Layers to freeze	6	5	5	1	1	0	0
Average validation set accuracy	30.8	38.1	42.7	46.6	52.2	56.3	62.1

For our smallest sized dataset using the best transfer learning settings compared to those often adopted from [19] results in a 47.88% increase in accuracy. The settings used to produce the best accuracy for each target dataset size are shown in Table 2.

4 Discussion

To our knowledge we have performed the most thorough analysis of transfer learning using CNNs to date. Based on this research we identified optimal transfer learning protocols for target dataset sizes from very small to large, when source and target datasets and tasks are similar. Our results show:

1. Using the commonly adopted transfer learning practice of transferring all but the final classification layer and training for 30 epochs at a learning rate of 0.01 causes a significant increase in overfitting compared to using optimal transfer learning protocols, particularly as the size of the target dataset decreases.
2. Best practice transfer learning and fine-tuning protocols for smaller target datasets cannot be derived using the results of the commonly adopted protocols used in performing transfer learning with a large target dataset.
3. There is an interaction between the optimal number of layers to transfer and fine-tuning hyperparameters. If this is not taken into account, the optimal number of layers to transfer will be overestimated for small and medium sized target datasets and underestimated for large target datasets.
4. Freezing layers of pre-trained weights after transferring them rarely results in the best results. It is only worth considering for very small target datasets.
5. Using our identified best practice transfer learning and fine-tuning protocols produces large changes in results compared to the existing protocols. This result again is particularly pronounced when applied to smaller datasets.

These new findings are likely to impact results of past and current research, which used commonly adopted protocols.

5 Conclusion

We have shown that using our identified optimal protocols when performing transfer learning results in a significant increase in accuracy when compared to the commonly adopted protocols. This is particularly evident for small target dataset sizes where we observed a 47.88% increase in accuracy compared to using the more standard protocols. Best practice transfer learning protocols for small target datasets cannot be predicted from previous research that has analysed transfer learning under very specific conditions that are not universally applicable. We have presented optimal transfer learning protocols for various target dataset sizes, from very small to large, when source and target datasets and tasks are similar. We intend to extend our work to analyse more diverse datasets and tasks, along with looking at ways to predict optimal transfer learning protocols. This paper and our ongoing research are significant as it is likely to improve outcomes from past, current and future research in transfer learning.

Acknowledgement. We thank Dawn Olley for her invaluable editing advice.

This work was supported by computational resources provided by the Australian Government through the National Computational Infrastructure (NCI) facility under the ANU Merit Allocation Scheme.

References

1. Agrawal, P., Girshick, R., Malik, J.: Analyzing the performance of multilayer neural networks for object recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 329–344. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10584-0_22

2. Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: Factors of transferability for a generic convnet representation. IEEE Trans. Pattern Anal. Mach. Intell. **38**(9), 1790–1802 (2016)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR09 (2009)
4. Donahue, J., et al.: Decaf: a deep convolutional activation feature for generic visual recognition. In: International Conference on Machine Learning, pp. 647–655 (2014)
5. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
6. He, K., Girshick, R., Dollár, P.: Rethinking imagenet pre-training. arXiv preprint arXiv:1811.08883 (2018)
7. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
8. Huh, M., Agrawal, P., Efros, A.A.: What makes ImageNet good for transfer learning? arXiv preprint arXiv:1608.08614 (2016)
9. Kornblith, S., Shlens, J., Le, Q.V.: Do better ImageNet models transfer better? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2661–2671 (2019)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
11. Mahajan, D., et al.: Exploring the limits of weakly supervised pretraining. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 181–196 (2018)
12. Mormont, R., Geurts, P., Marée, R.: Comparison of deep transfer learning strategies for digital pathology. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 2262–2271 (2018)
13. Paszke, A., et al.: Automatic differentiation in PyTorch. In: NIPS Autodiff Workshop (2017)
14. Scott, T., Ridgeway, K., Mozer, M.C.: Adapted deep embeddings: a synthesis of methods for k-shot inductive transfer learning. In: Advances in Neural Information Processing Systems, pp. 76–85 (2018)
15. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: 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)
16. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 843–852 (2017)
17. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019)
18. Wu, Y., Hassner, T., Kim, K., Medioni, G., Natarajan, P.: Facial landmark detection with tweaked convolutional neural networks. IEEE Trans. Pattern Anal. Mach. Intell. **40**(12), 3067–3074 (2018)
19. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, pp. 3320–3328 (2014)