BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Transfer Learning for Deep Learning Applications: Methods, Problems, and a Way to Evaluate Them

Aditya Ajay Deshmane

Dept. of Artificial Intelligence and Data Science
Ajeenkya DY Patil School of Engineering, Pune, Maharashtra, India
Email : aadityadeshmane.in@gmail.com

Dr. Vishal Puranik

Dept. of Artificial Intelligence and Data Science
Ajeenkya DY Patil School of Engineering, Pune, Maharashtra, India
Email : vishalpura@gmail.com

Abstract

Deep learning models perform well on image, text and speech problems. The catch is that they normally require a large labelled dataset and powerful hardware, and many small projects lack both. Transfer learning is one way to handle this situation: a model that has already been trained on one task is trained further on a smaller task of one's own. In this paper we review transfer learning for deep learning applications. The work we studied falls into four groups: using a model to extract features, fine-tuning a model, adapting the domain, and changing only a small part of a big model. We also describe the main problems, namely negative transfer, a mismatch between source and target data, overfitting when the dataset is small, and deciding which layers to update. From the review we saw that transfer learning helps a lot. However, the studies are hard to compare, because each uses its own datasets, data sizes and metrics. For this reason we propose a pipeline together with an evaluation plan. The plan looks at four things at once: accuracy, the amount of data needed, training cost, and consistency across repeated runs. This is a pre-implementation paper. The results table is left empty for now and will be filled after the experiments are done.

Keywords: Transfer Learning, Deep Learning, Pre-trained Models, Fine-Tuning, Domain Adaptation, Negative Transfer, Small Datasets

Introduction

Deep learning is used today for image recognition, language understanding, speech and many other tasks. Deep learning comes with a price, and that price is data. A large network trained from scratch may need hundreds of thousands of labelled examples. Getting those examples is slow and costly. In fields such as medical imaging and farming, experts are needed for the labelling. GPUs are also costly, and many students and small teams cannot get them easily.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Transfer learning reduces the need for training by reusing work that has already been done. In transfer learning, a network is first trained on a dataset and the learned weights of the network can become the starting point for a new related problem (Pan & Yang, 2010). For example, one can start with a network that has been trained on a collection of images and then train that same network further on a small personal set of images.

Experiments on the steps of networks reveal that the first layers of the network learn general patterns, while the last layers depend more on the original task (Yosinski et al., 2014). This discovery allows us to keep the early layers of the network unchanged and retrain only the remaining parts. Many researchers have worked on this topic. The first surveys explained the basic types of transfer (Pan & Yang, 2010), and newer surveys explain the methods (Zhuang et al., 2021). Recent papers focus on very large pre-trained models and try to adapt them without changing every parameter (Houlsby et al., 2019; Hu et al., 2022).

Reuse can also backfire. If the new problem is far from the one the model came from, starting with the old weights may leave us worse off than starting from random weights, an effect called negative transfer (Wang et al., 2019). A pre-trained model is therefore a choice that has to be justified. Three decisions are left to us: the model to begin with, how deep the retraining should go, and a scratch run to compare against.

In this paper we look at transfer learning as a problem of choosing methods and testing them properly. Instead of repeating the surveys that already exist, we try to show how the main methods connect, which choices a beginner has to make, and how a study should be evaluated with the results.

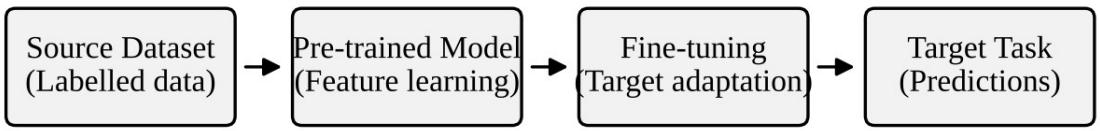


Figure 1: General workflow of transfer learning.

Motivation and Objectives

This work began with a real-world problem. Training a model from scratch is difficult when there is limited data, and small groups need a way to build a good model. Another problem is that papers about transfer learning show their results in different ways, so it is not clear which method to choose for a specific problem.

The objectives of this study are:

12. To explain how knowledge moves from one trained network to another, and why this knowledge works in networks.
13. To compare feature extraction, fine-tuning, domain adaptation and methods that train only a few parameters.
14. To determine when transfer learning is useful and when it is not.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

15. To list the main problems: negative transfer, overfitting and domain shift.

16. To come up with a process, and a way to check how well it works.

We concentrate on finding answers to the following research questions.

RQ1: How do feature extraction, partial fine-tuning, and full fine-tuning differ in accuracy and cost?

RQ2: Does the amount of target data change which method is best?

RQ3: How does the difference between source and target data affect transfer, and can negative transfer be spotted early?

RQ4: Can methods that train only a few parameters come close to full fine-tuning at a lower cost?

RQ5: What is a good way to evaluate results so that we do not rely on one best score?

Technical Foundations

Basic Idea

Pan and Yang (2010) gave names to the settings based on what changes between the original training and the new problem. Imagine we have a network that was trained on ImageNet. If we use it to classify skin lesions, the task itself is new; this is their inductive case. If we use it for the same task but with pictures from another camera, the task remains the same but the input data changes, and this is the transductive case. If those new pictures also do not have any labels, we are in the unsupervised case.

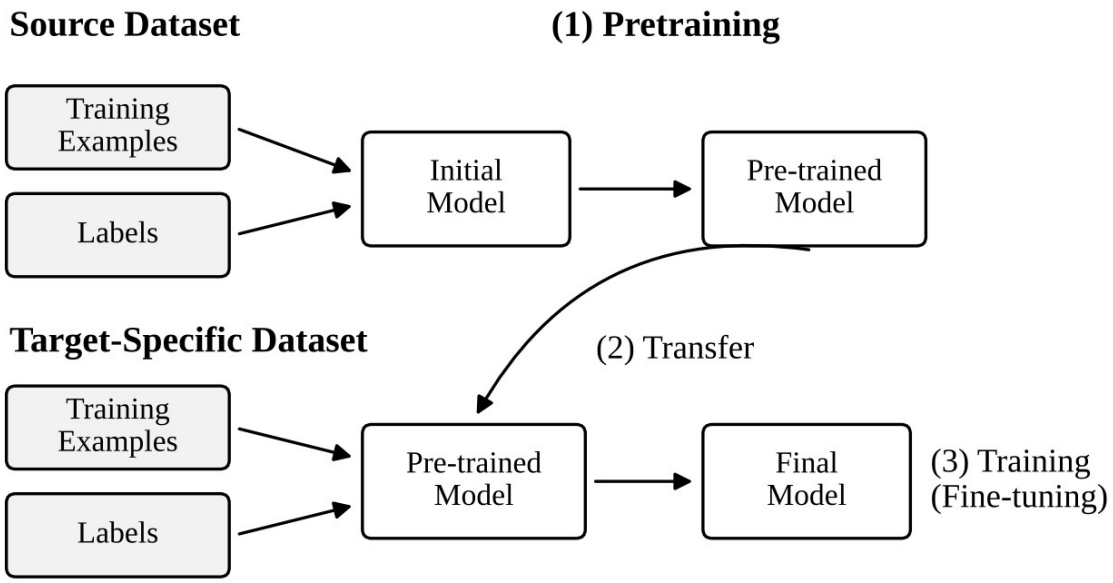


Figure 2: Transfer learning workflow showing pretraining on a source dataset, transfer of learned knowledge, and fine-tuning on a target-specific dataset.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Pre-trained Models

In image problems the starting point is usually a network that was trained on ImageNet. In language problems the starting point is a model that has already read a large text collection and is then tuned for one job, such as classifying reviews (Howard & Ruder, 2018). It is tempting to think that the strongest source model always gives the best result, but Kornblith et al. (2019) found exceptions on some target datasets.

Feature Extraction and Fine-Tuning

The easiest way to use a pre-trained network is to keep it frozen. Its output is given to a new classifier and only that classifier is trained. This works quickly and requires little data, so we see it as the first starting point. Because the network stays the same, its features do not change for the task.

Fine-tuning means taking a model that has already been trained and adapting it to a new task. We can train all of the layers or just the last few layers. Yosinski et al. (2014) showed that fine-tuning can improve transfer, mainly when the source and target are not very close. Howard and Ruder (2018) unfroze the layers one at a time and used different learning rates for different layers so that the model does not lose what it already knows. Careless fine-tuning can also wipe out useful source knowledge.

Domain Adaptation

Sometimes the source data and the target data look different, for example photos from two cameras. Domain adaptation tries to make the features similar for both. Ganin et al. (2016) used an adversarial setup for this: the network learns features that help with the task but give no clue about which domain an input came from.

Training Only a Few Parameters

Large pre-trained models have millions or billions of parameters. Because of this, training them from scratch for each task uses a lot of time and computing power. Houlsby et al. (2019) placed small trainable modules, called adapters, inside a frozen network. Their results were almost as good as fine-tuning, but only a small number of the weights were updated. LoRA uses a similar approach: it adds trainable low-rank matrices to certain layers and keeps the original weights frozen (Hu et al., 2022).

Negative Transfer

Negative transfer occurs when knowledge from one task makes learning a new task harder. It is more likely when the two domains are far apart, when the target data is noisy, or when too much of the source model is forced onto the target task. Because of this, an experiment should always include a model trained from scratch to compare with.

Literature Review

Pan and Yang (2010) wrote a survey that is often cited. The survey explains transfer learning and separates it into categories based on how the source and target domains and tasks relate. A later survey by Zhuang et al. (2021) covers more transfer learning methods and shows where transfer learning is applied.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

To find out why transfer works in networks, Yosinski et al. (2014) tested how well the features from each layer can be moved to a new task. Some features did not work well after transfer, but fine-tuning helped improve the results. Kornblith et al. (2019) tried ImageNet models on several target datasets. In most cases the models with higher ImageNet accuracy also transferred better, though there were exceptions.

For language models, Howard and Ruder (2018) proposed a fine-tuning method that uses different learning rates and slow unfreezing. With this method, a pre-trained language model can handle text classification using a small amount of labelled data, showing that language models can be effective even with limited data.

Ganin et al. (2016) focused on domain adaptation and took on learning features that help the task but stay alike across domains. Wang et al. (2019) examined negative transfer directly and offered a method to cut down the harmful impact of source knowledge.

Medical images are an example to consider because the data is not very plentiful and the images look different from regular pictures. Raghu et al. (2019) found that using transfer from ImageNet provided small improvements in the medical cases they looked at, and that small and simple models performed about the same way. Other work, like adapters (Houlsby et al., 2019) and LoRA (Hu et al., 2022), tries to lower the cost of adapting models.

The papers overall show that transfer learning can be very effective in many situations. How much it improves results depends on the area you are working in, the target domain, how much data you have available, and the method that is used.

Literature Comparison and Research Gap

Table 1 gives a short comparison of the studies we reviewed. From the review we found four gaps.

Table 1: Comparison of Representative Studies

Study	Focus	Main Contribution	Limitation / Gap
Pan and Yang	Types of transfer learning	Divided transfer learning into inductive, transductive, and unsupervised	Written before deep learning became common
Yosinski et al.	How features transfer	Showed how general or specific the features of each layer are	Tested on a small set of image tasks
Zhuang et al.	Large survey	Covered many transfer methods and uses	Only a survey; no common experiment
Howard and Ruder	Fine-tuning for text	Gradual unfreezing and different learning rates per layer	Mostly for text classification
Ganin et al.	Domain adaptation	Adversarial learning of features that work across domains	Training can be unstable

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Study	Focus	Main Contribution	Limitation / Gap
Houlsby et al.	Adapters	Small trainable modules in a frozen model	Adds extra layers and settings to tune
Hu et al.	LoRA	Updates with very few trainable weights	Rank and position need tuning
Kornblith et al.	ImageNet transfer	Checked if better source models transfer better	Not true for all target datasets
Raghu et al.	Medical images	Questioned the benefit of normal ImageNet transfer	Limited to the datasets they used
Wang et al.	Negative transfer	A method to reduce harmful source knowledge	Needs a way to measure how useful the source is

1. **Incomplete reporting:** Many studies only give the accuracy. Training time, memory use, number of trained parameters and the effect of data size are often left out. These details decide whether a method is practical.
2. **Negative transfer is rarely measured:** Papers mention it, but a beginner often has no simple method to verify whether a chosen model fits the data, and may feel uncertain about the model and the data it should work with.
3. **Single runs:** Training involves some randomness. A gain of one or two points might disappear when the experiment is run again with different seeds.
4. **Special domains:** Normal pre-training may help less when the data looks very different from the source, as with medical images, where pre-training is not as effective (Raghu et al., 2019).

So the gap we see is the lack of a repeatable way to compare transfer methods on accuracy, data needed, cost and stability together. We do not claim any algorithm without experiments to support it.

Proposed System

The pipeline we suggest has simple parts. First we prepare the data. Then we choose a pre-trained model. After that we decide how to adapt it. Next we do the training. Finally we do the evaluation. This way we build a system step by step. Figure 3 shows the steps.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

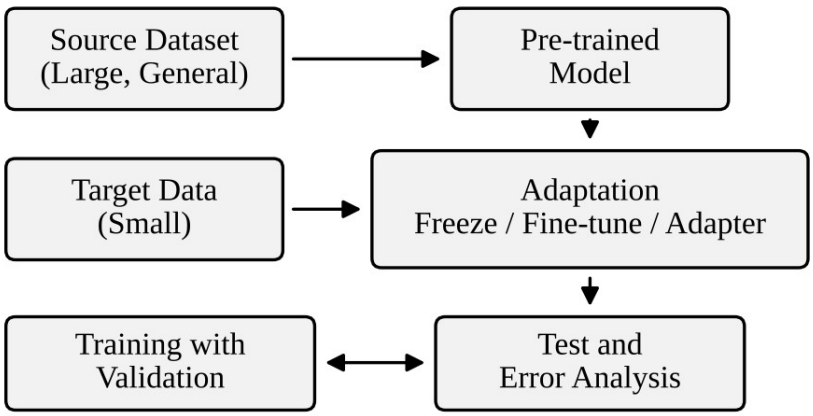


Figure 3: Proposed transfer learning pipeline.

Data Preparation

Before any training starts, the target data is split into three parts. One part is used to train the model. Another part is used to adjust the settings. The third part is saved for the test, and we do not look at this part until the end. For images, preparation can involve resizing, making the data consistent and simple changes. For text, we use the tokenizer that is included with the pre-trained model we have selected. The data should match what the model may see in real situations.

Model and Adaptation Choices

We choose one or two models of different sizes. Each pre-trained model is adapted in different ways, and these adaptations are compared:

- a new model trained from the start, to compare its results with other models;
- feature extraction with a frozen model;
- fine-tuning only the last layers;
- fine-tuning the whole model;
- adapter or LoRA training.

We add the pieces one by one so that the result of each decision is easy to notice.

Design Rules

We follow these rules in the pipeline:

- Every study has a scratch baseline. Every study also has a frozen-feature baseline.
- Settings are chosen using the validation data only.
- Pre-trained layers get a smaller learning rate than new layers.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

- In full fine-tuning, layers are unfrozen slowly.
- Early stopping and regularization are used to reduce overfitting.
- Every setup is run with several random seeds.

Experimental Methodology

We want the experiments to show when transfer learning helps, how much transfer learning helps, and what transfer learning costs.

Datasets

We plan to use one or two datasets from different areas: for example, one general image dataset and one special dataset, such as plant diseases or chest X-rays. We should note the source, licence, size and class balance for each dataset. To see how much data is needed, each method is trained on parts of the target training data, such as 10%, 25%, 50%, and 100%, because data quantity is important for good results.

Steps of the Experiment

The experiment goes through these steps:

1. choose and clean the datasets;
2. split into training, validation, and test sets;
3. train the scratch baseline;
4. feature extraction with a frozen model;
5. partial and full fine-tuning;
6. adapter or LoRA training;
7. repeat with different random seeds;
8. ablation tests and error analysis.

Reproducibility

We should write down the model name and version, the learning rate, the batch size, the optimizer, the number of epochs, the augmentation, the random seed and the hardware. Results change a little from one run to the next, so we repeat each setup and provide the average along with its spread.

Evaluation Metrics

For prediction quality we use accuracy, precision, recall and F1-score. We show a confusion matrix for the final models. When one class has more samples than the other classes, macro-F1 or AUC gives a fairer picture than accuracy alone.

Besides prediction quality, we keep track of how long training takes, how many parameters are actually being trained, the memory used and the time it takes for each prediction. We call transfer benefit the

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

improvement compared to the scratch baseline. If it is negative, that means negative transfer has occurred.

Robustness can be checked by using data that has noise or tiny changes. A single number should not be used to say that one method is better than another.

The approach also has limits. The result depends on the pre-trained model, and any bias or weakness in that model can pass to the task. The result changes with the settings and with the way the data is split. Public datasets may not match real use, and large models need computing power that may not be available. For these reasons, a model that does well on a benchmark should be checked again on data close to its real use before it is trusted.

Planned Ablations

The things we are going to test are: how many layers we freeze, what learning rate we use for the pre-trained layers compared to the new layers, how using data augmentation affects things, what size and kind of source model we use, and what size of the adapter or rank of LoRA we use.

Tools

We can do the work using Python with PyTorch or TensorFlow. Pre-trained models are available from Hugging Face or the timm library. Scikit-learn helps measure the metrics. Google Colab or Kaggle provide GPU access. Experiments can be tracked with TensorBoard or Weights & Biases. The code can be stored in Git.

Results Analysis and Discussion

The final version of this paper must include results that were actually measured during the experiments. It should not have guesses or assumed numbers. Table 2 explains how the results will be shown.

Table 2: Results table to be completed after experiments

Configuration	Acc.	F1	Time	Params
From scratch	TBD	TBD	TBD	TBD
Feature extraction	TBD	TBD	TBD	TBD
Partial fine-tuning	TBD	TBD	TBD	TBD
Full fine-tuning	TBD	TBD	TBD	TBD
Adapter / LoRA	TBD	TBD	TBD	TBD

The main comparison is between training from scratch and each transfer method, for every part of the target data. When a transfer method is better, we should check that the gain stays about the same over runs and is not just luck.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Cost should be checked well. A method that gives a gain but needs a large amount of time or memory may not be useful. If a pre-trained model does worse than the scratch model, we should look at it carefully because it may be negative transfer.

The wrong predictions should be looked at one class at a time to see where the model is failing. It is important to check whether the mistakes happen because the source data and target data are different. This helps understand the model and fix the problems.

No result should be written in the paper until the experiment has really been done.

Applications and Limitations

Transfer learning is used in many areas. In medicine, transfer learning helps find diseases in X-rays or CT scans when only a little labelled data is available. In farming, it helps identify crop and leaf diseases from photos. In language tasks, it makes text classification and question answering possible with small labelled sets. Transfer learning is also used for speech recognition in languages with few recordings, for finding defects in factories, and for mapping land use from satellite images.

Future Scope

In the future, work can be done on methods to predict how well a source model will perform when transferred, before any training begins. This would help prevent poor performance. Improving how models handle domain shifts, and transfer learning for small datasets, are also important areas to explore. Adapting models with less memory can make transfer learning work on smaller computers. Another path is to look at how bias in the source model ends up in the new model. Being able to explain the predictions is also very important in areas like healthcare, where things matter a lot.

Conclusion

In this paper we reviewed transfer learning for deep learning and suggested a simple pipeline and evaluation plan for comparing different ways of adapting models. Transfer learning makes deep learning practical when data and computing power are limited. The papers we studied show progress from feature extraction and fine-tuning to domain adaptation and methods that train only a few parameters. The review also shows that using transfer learning does not always work well, and that comparisons between methods are often not complete. Our plan looks at accuracy, the amount of data needed, the cost and the stability at the same time, and it always compares with a model that was trained without any prior learning.

Experiments in the next stage will reveal which methods work best and under what conditions. Any claim of superiority must come only after those experiments are finished.

References

36. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. Journal of Machine Learning Research, 17(59), 1–35.
37. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). Parameter-efficient transfer learning for NLP. Proceedings of the 36th International Conference on Machine Learning (ICML), PMLR 97, 2790–2799.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

38. Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), 328–339.
39. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. International Conference on Learning Representations (ICLR).
40. Kornblith, S., Shlens, J., & Le, Q. V. (2019). Do better ImageNet models transfer better? Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2661–2671.
41. Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191
42. Raghu, M., Zhang, C., Kleinberg, J., & Bengio, S. (2019). Transfusion: Understanding transfer learning for medical imaging. Advances in Neural Information Processing Systems (NeurIPS), 32.
43. Wang, Z., Dai, Z., Póczos, B., & Carbonell, J. (2019). Characterizing and avoiding negative transfer. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11293–11302.
44. Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? Advances in Neural Information Processing Systems (NeurIPS), 27.
45. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555