

Get started

Open in app



Follow

588K Followers



You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)

DEEP LEARNING CASE STUDIES

4 Pre-Trained CNN Models to Use for Computer Vision with Transfer Learning

Using State-of-the-Art Pre-trained Neural Network Models to Tackle Computer Vision Problems with Transfer Learning



Orhan G. Yalçın · Sep 23, 2020 · 8 min read ★

Before we start, if you are reading this article, I am sure that we share similar interests and are/will be in similar industries. So let's connect via [Linkedin](#)! Please do not hesitate to send a contact request! [Orhan G. Yalçın — Linkedin](#)

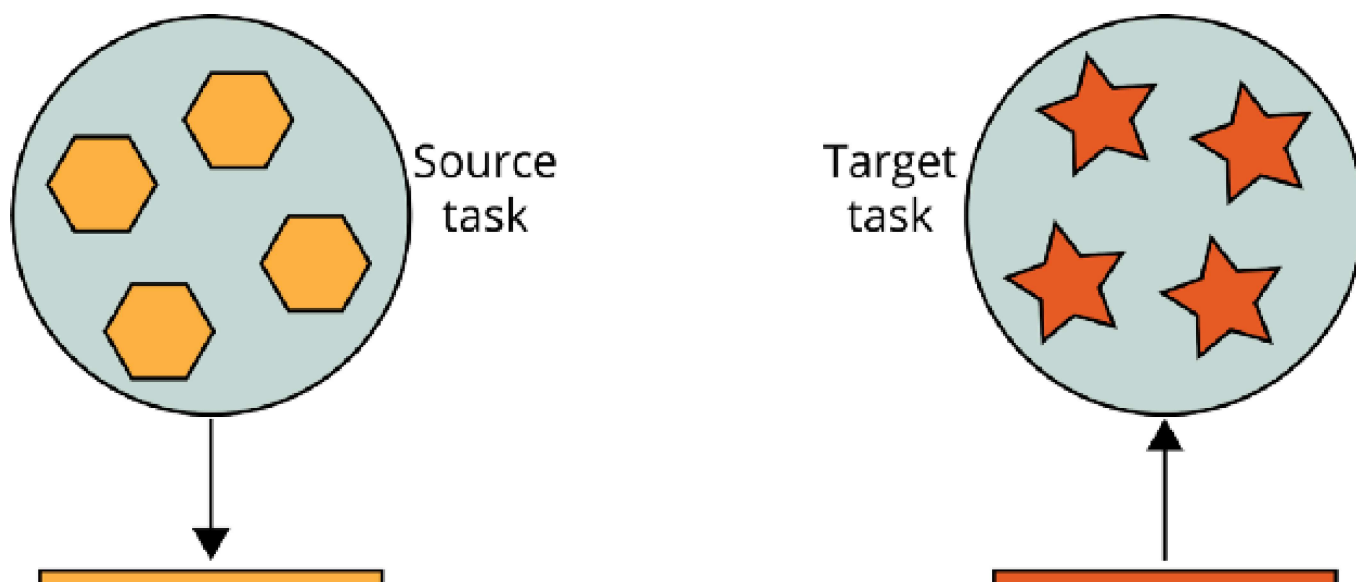




Figure 1. How Transfer Learning Works (Image by Author)

If you have been trying to build machine learning models with high accuracy; but never tried Transfer Learning, this article will change your life. At least, it did mine!

Most of us have already tried several machine learning tutorials to grasp the basics of neural networks. These tutorials were very helpful to understand the basics of artificial neural networks such as Recurrent Neural Networks, Convolutional Neural Networks, GANs, and Autoencoders. But, their main functionality was to prepare you for real-world implementations.

Now, if you are planning to build an AI system that utilizes deep learning, you either (i) have to have a very large budget for training and excellent AI researchers at your disposal or (ii) have to benefit from transfer learning.

What is Transfer Learning?

Transfer learning is a subfield of machine learning and artificial intelligence which aims to apply the knowledge gained from one task (source task) to a different but similar task (target task).

For example, the knowledge gained while learning to classify Wikipedia texts can be used to tackle legal text classification problems. Another example would be using the knowledge gained while learning to classify cars to recognize the birds in the sky. As you can see there is a relation between these examples. We are not using a text classification model on bird detection.

Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned.

In summary, transfer learning is a field that *saves you from having to reinvent the wheel* and helps you build AI applications in a very short amount of time.



Figure 2. Don't Reinvent the Wheel, Transfer the Existing Knowledge (Photo by [Jon Cartagena](#) on [Unsplash](#))

History of Transfer Learning

To show the power of transfer learning, we can quote from [Andrew Ng](#):

Transfer learning will be the next driver of machine learning's commercial success after supervised learning.

The history of Transfer Learning dates back to 1993. With her paper, [Discriminability-Based Transfer between Neural Networks](#), Lorien Pratt opened the pandora's box and introduced the world with the potential of transfer learning. In July 1997, the journal

Machine Learning published a special issue for transfer learning papers. As the field advanced, adjacent topics such as multi-task learning were also included under the field of transfer learning. Learning to Learn is one of the pioneer books in this field. Today, transfer learning is a powerful source for tech entrepreneurs to build new AI solutions and researchers to push the frontiers of machine learning.

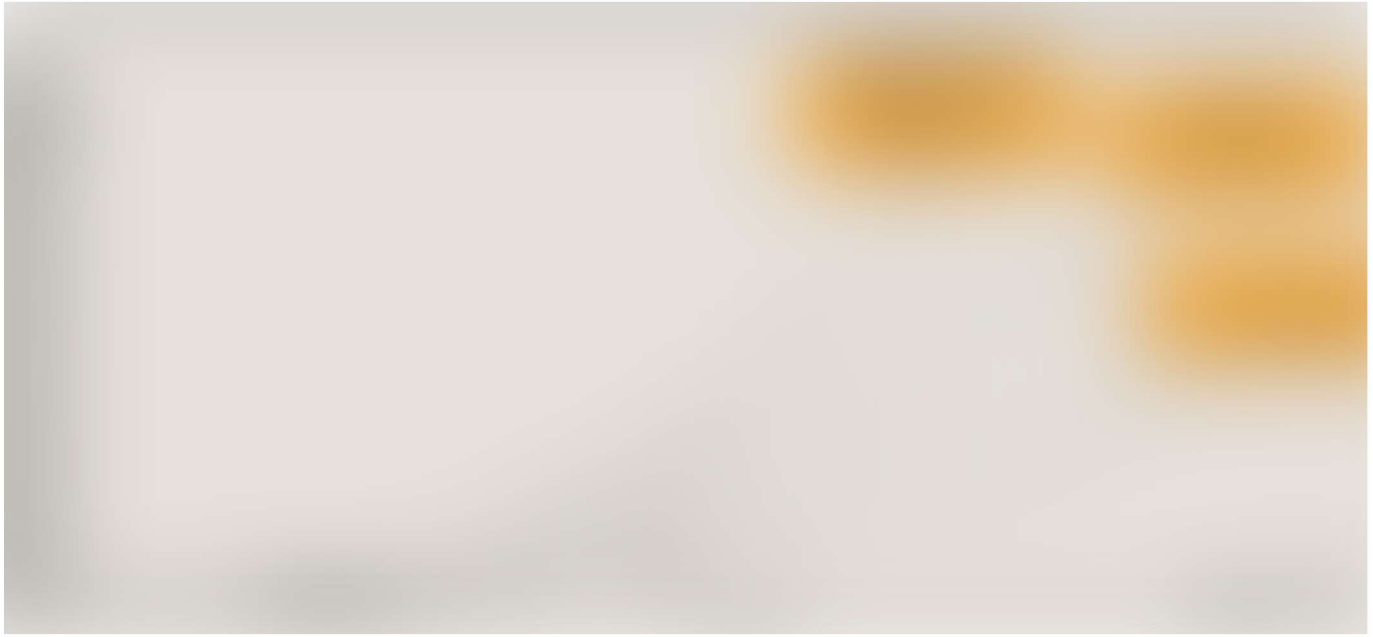


Figure 3. Andrew Ng's Expectation for the Commercial Success of Machine Learning Subfields (Image by Author)

How Does Transfer Learning Work?

There are three requirements to achieve transfer learning:

- Development of an Open Source Pre-trained Model by a Third Party
- Repurposing the Model
- Fine Tuning for the Problem

Development of an Open Source Pre-trained Model

A pre-trained model is a model created and trained by someone else to solve a problem that is similar to ours. In practice, someone is almost always a tech giant or a group of star researchers. They usually choose a very large dataset as their base datasets such as ImageNet or the Wikipedia Corpus. Then, they create a large neural network (e.g., VGG19 has 143,667,240 parameters) to solve a particular problem (e.g., this problem is

image classification for VGG19). Of course, this pre-trained model must be made public so that we can take these models and repurpose them.

Repurposing the Model

After getting our hands on these pre-trained models, we repurpose the learned knowledge, which includes the layers, features, weights, and biases. There are several ways to load a pre-trained model into our environment. In the end, it is just a file/folder which contains the relevant information. However, deep learning libraries already host many of these pre-trained models, which makes them more accessible and convenient:

- [TensorFlow Hub](#)
- [Keras Applications](#)
- [PyTorch Hub](#)

You can use one of the sources above to load a trained model. It will usually come with all the layers and weights and you can edit the network as you wish.

Fine-Tuning for the Problem

Well, while the current model may work for our problem. It is often better to fine-tune the pre-trained model for two reasons:

- So that we can achieve even higher accuracy;
- Our fine-tuned model can generate the output in the correct format.

Generally speaking, in a neural network, while the bottom and mid-level layers usually represent general features, the top layers represent the problem-specific features. Since our new problem is different than the original problem, we tend to drop the top layers. By adding layers specific to our problems, we can achieve higher accuracy.

After dropping the top layers, we need to place our own layers so that we can get the output we want. For example, a model trained with ImageNet can classify up to 1000 objects. If we are trying to classify handwritten digits (e.g., MNIST classification), it may be better to end up with a final layer with only 10 neurons.

After we add our custom layers to the pre-trained model, we can configure it with special loss functions and optimizers and fine-tune with extra training.

For a quick Transfer Learning tutorial, you may visit the post below:

Fast Neural Style Transfer in 5 Minutes with TensorFlow Hub & Magenta

Transferring the van Gogh's Unique Style to Photos with Magenta's Arbitrary Image Stylization Network and Deep Learning

towardsdatascience.com

4 Pre-Trained Models for Computer Vision

Here are the four pre-trained networks you can use for computer vision tasks such as ranging from image generation, neural style transfer, image classification, image captioning, anomaly detection, and so on:

- VGG19
- Inceptionv3 (GoogLeNet)
- ResNet50
- EfficientNet

Let's dive into them one-by-one.

VGG-19

VGG is a convolutional neural network which has a depth of 19 layers. It was build and trained by Karen Simonyan and Andrew Zisserman at the University of Oxford in 2014 and you can access all the information from their paper, [Very Deep Convolutional Networks for Large-Scale Image Recognition](#), which was published in 2015. The VGG-19 network is also trained using more than 1 million images from the ImageNet database. Naturally, you can import the model with the ImageNet trained weights. This pre-trained network can classify up to 1000 objects. The network was trained on 224x224 pixels colored images. Here is brief info about its size and performance:

- **Size:** 549 MB

- **Top-1:** Accuracy: 71.3%
- **Top-5:** Accuracy: 90.0%
- **Number of Parameters:** 143,667,240
- **Depth:** 26



Figure 4. An Illustration of the VGG-19 Network (Figure by [Clifford K. Yang](#) and [Yufeng Zheng](#) on [ResearchGate](#))

Inceptionv3 (GoogLeNet)

Inceptionv3 is a convolutional neural network which has a depth of 50 layers. It was build and trained by Google and you can access all the information on the paper, titled “[Going deeper with convolutions](#)”. The pre-trained version of Inceptionv3 with the ImageNet weights can classify up to 1000 objects. The image input size of this network was 299x299 pixels, which is larger than the VGG19 network. While VGG19 was the runner up in 2014’s ImageNet competition, Inception was the winner. The brief summary of Inceptionv3 features is as follows:

- **Size:** 92 MB
- **Top-1:** Accuracy: 77.9%
- **Top-5:** Accuracy: 93.7%

- **Number of Parameters:** 23,851,784
- **Depth:** 159

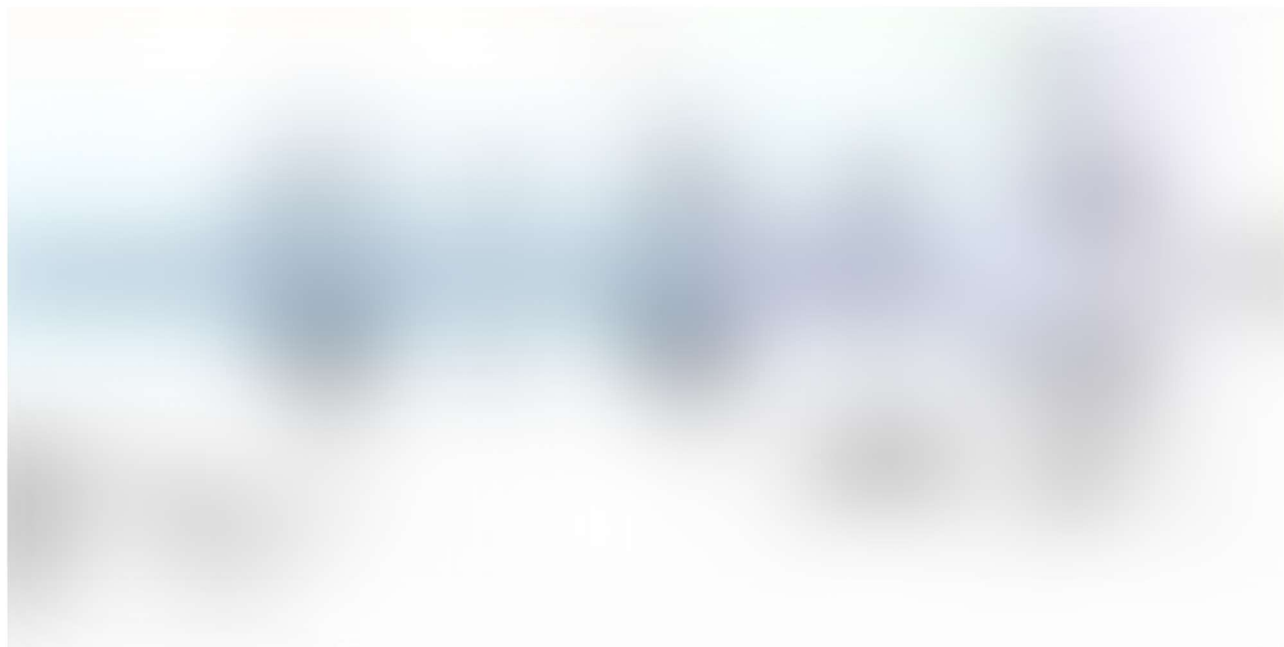


Figure 5. An Illustration of the Inceptionv3 Network (Figure by [Masoud Mahdianpari](#), [Bahram Salehi](#), and [Mohammad Rezaee](#) on [ResearchGate](#))

ResNet50 (Residual Network)

ResNet50 is a convolutional neural network which has a depth of 50 layers. It was build and trained by Microsoft in 2015 and you can access the model performance results on their paper, titled [Deep Residual Learning for Image Recognition](#). This model is also trained on more than 1 million images from the ImageNet database. Just like VGG-19, it can classify up to 1000 objects and the network was trained on 224x224 pixels colored images. Here is brief info about its size and performance:

- **Size:** 98 MB
- **Top-1:** Accuracy: 74.9%
- **Top-5:** Accuracy: 92.1%
- **Number of Parameters:** 25,636,712

If you compare ResNet50 to VGG19, you will see that ResNet50 actually outperforms VGG19 even though it has lower complexity. ResNet50 was improved several times and

you also have access to newer versions such as [ResNet101](#), [ResNet152](#), [ResNet50V2](#), [ResNet101V2](#), [ResNet152V2](#).



Figure 6. An Illustration of the ResNet50 Network (Figure by [Masoud Mahdianpari](#), [Bahram Salehi](#), and [Mohammad Rezaee](#) on [ResearchGate](#))

EfficientNet

EfficientNet is a state-of-the-art convolutional neural network that was trained and released to the public by Google with the paper “[EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks](#)” in 2019. There are 8 alternative implementations of EfficientNet (B0 to B7) and even the simplest one, EfficientNetB0, is outstanding. With 5.3 million parameters, it achieves a 77.1% Top-1 accuracy performance.





Figure 7. EfficientNet Model Size vs. ImageNet Accuracy (Plot by [Mingxing Tan](#) and [Quoc V. Le](#) on [Arxiv](#))

The brief summary of EfficientNetB0 features is as follows:

- **Size:** 29 MB
- **Top-1:** Accuracy: 77.1%
- **Top-5:** Accuracy: 93.3%
- **Number of Parameters:** ~5,300,000
- **Depth:** 159

Other Pre-Trained Models for Computer Vision Problems

We listed the four state-of-the-art award-winning convolutional neural network models. However, there are dozens of other models available for transfer learning. Here is a benchmark analysis of these models, which are all available in [Keras Applications](#).

Table 1. Benchmark Analysis of Pre-Trained CNN Models (Table by Author)

Conclusion

In a world where we have easy access to state-of-the-art neural network models, trying to build your own model with limited resources is like trying to reinvent the wheel. It is pointless.

Instead try to work with these train models, add a couple of new layers on top considering your particular computer vision task, and train. The results will be much more successful than a model you build from scratch.

Subscribe to the Mailing List for the Full Code

If you would like to have access to full code on Google Colab and have access to my latest content, subscribe to the mailing list: 

Subscribe Now

Enjoyed the Article?

If you like this article, consider checking out my other similar articles:

Image Classification in 10 Minutes with MNIST Dataset

towardsdatascience.com

TensorFlow and VGG19 Can Help You Convert Your Photos into Beautiful Pop Art Pieces

Neural Style Transfer based on Andy Warhol's Monroe Diptych with pre-trained computer vision network VGG19, transfer...

towardsdatascience.com

Image Generation in 10 Minutes with Generative Adversarial Networks

Using Unsupervised Deep Learning to Generate Handwritten Digits with Deep Convolutional GANs using TensorFlow and the...

towardsdatascience.com

Sentiment Analysis in 10 Minutes with BERT and Hugging Face

Learn the basics of the pre-trained NLP model, BERT, and build a sentiment classifier using the IMDB movie reviews...

towardsdatascience.com

Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. [Take a look.](#)

Get this newsletter

You'll need to sign in or create an account to receive this newsletter.

[Transfer Learning](#)

[Artificial Intelligence](#)

[Machine Learning](#)

[Deep Learning](#)

[Computer Vision](#)

[About](#) [Help](#) [Legal](#)

Get the Medium app

