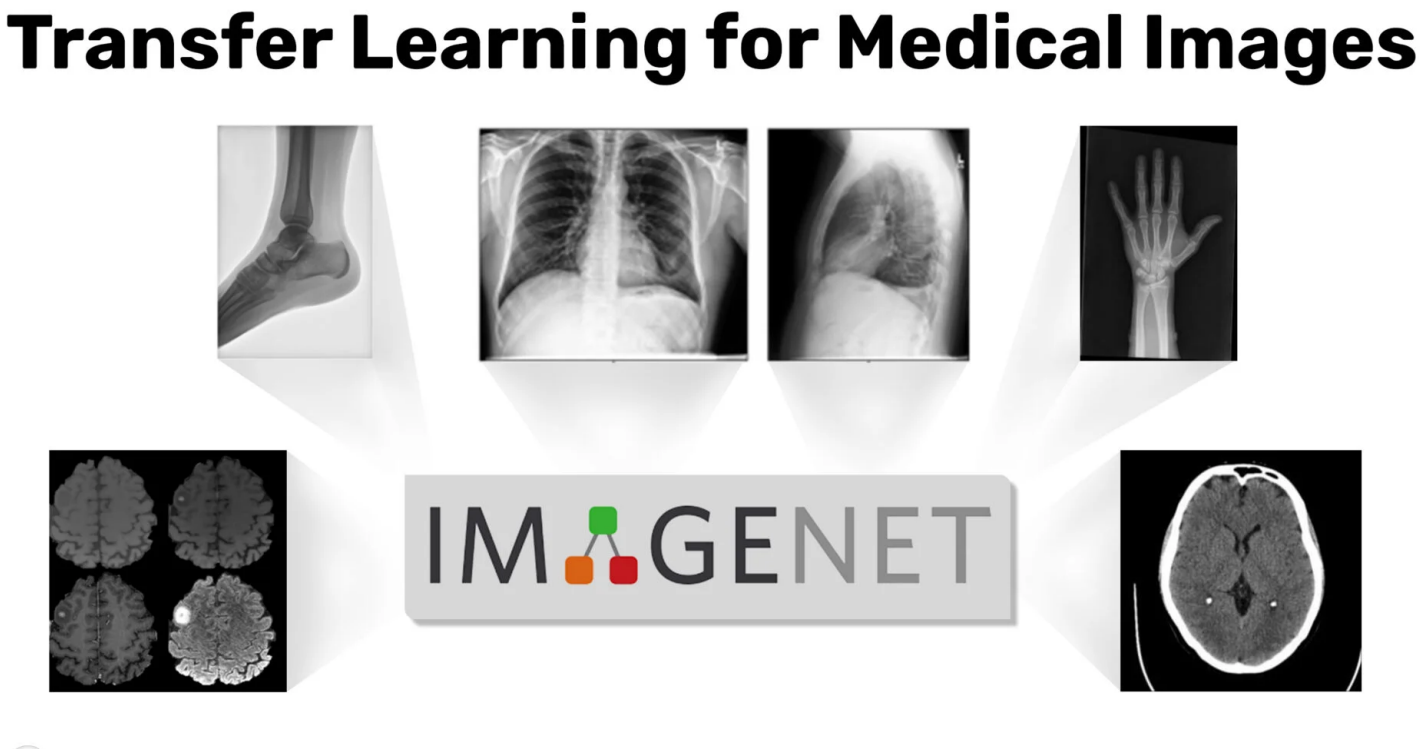
Vielen Dank, dass Sie Plastischer Reader ausprobieren. Teilen Sie uns Ihr Feedback mit.

Transfer Learning for Medical Images

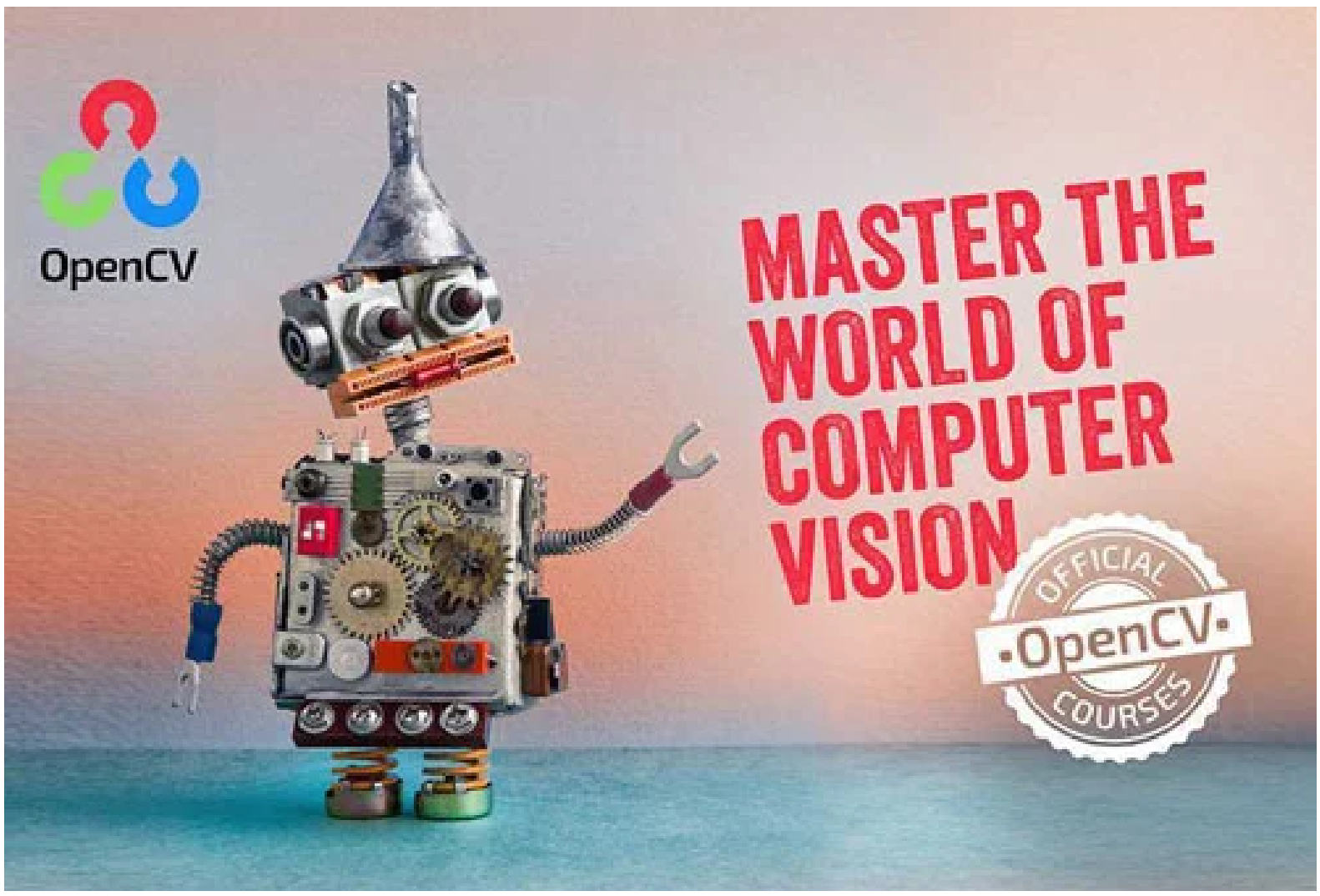


Our consulting company, Big Vision, has a long history of solving challenging computer vision and AI problems in diverse fields ranging from document analysis, security, manufacturing, real estate, beauty and fashion, automotive, and medical diagnostics, to name a few.

The spectacular growth of AI also means that the knowledge we acquired just a year back is now outdated. So, we continuously learn, and before embarking on a new problem, we do an extensive survey of state of the art in the industry.

We recently started working on a new medical diagnostic problem that utilizes X-ray photographs. This blog post distills the knowledge we refreshed about transfer Learning applied to medical data.

Before we dive into the details, let's quickly review the steps we use for training a new Deep learning model.



Official OpenCV Courses
Start your exciting journey from an absolute Beginner to Mastery in AI, Computer Vision & Deep Learning!
Learn More

1. Steps for Training a Deep Learning Model

Using Deep Learning, AI has made massive progress in solving computer vision problems like image classification, object detection, image segmentation, pose estimation, etc.

When we encounter a new problem, the steps for obtaining a reasonably good model are well-established.

1. **Prepare data:** Annotate Data, divide it into training, validation, and test sets.
2. **Choose a standard model** trained on ImageNet big enough to overfit the training set.
3. **Fine-Tuning / Transfer Learning**: Modify the network to fit your problem, and use transfer learning/fine-tuning to train the last few layers. Typically you start with ImageNet pre-trained weights.
4. **Check dataset:** Analyze errors. Check if the data is noisy or inconsistent. If needed, go back to step one and iterate a few times.
5. **Hyper-parameter optimization / Model architecture**: To squeeze out more accuracy, perform hyper-parameter optimization or even play with many different architectures.

This post will share our thoughts on steps 2 and 3 for medical data.

2. Should you use ImageNet architectures for solving medical problems?

Absolutely.

Creating a new model architecture is an expensive and time-consuming black art that we should leave to researchers in academia and industrial research labs.

Should you decide to go down that path, the gains in accuracy you see will be minuscule (if any). That is precisely why almost nobody starts with a new model architecture.

You can find many examples in this paper. Later in this post, we will share several more instances where human-level performance was achieved using standard ImageNet architectures.

There are always exceptions to this rule. For example, CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images uses a custom model.

3. Which ImageNet architectures are popular in the medical domain?

The following paper presents a large-scale review of transfer learning in medical image analysis.

A scoping review of transfer learning research on medical image analysis using ImageNet

The paper also mentions the widespread success of transfer learning using ImageNet pretrained weights for medical image analysis.

The table below shows popular model families used for various anatomical regions.

Anatomical Region	Model Architecture
Breast	Inception
Eyes	VGGNet
Skin	VGGNet
Tooth	VGGNet
Brain	AlexNet
Lungs	DenseNet

The table below shows popular model families used for various imaging modalities.

Imaging Modality	Model Architecture
Ultrasound	Inception
Endoscopy	Inception
Skeletal system X-rays	Inception
Fundus	VGGNet
Optical Coherence Tomography (OCT)	VGGNet
Brain MRI	AlexNet
Breast X-Rays	AlexNet

4. Should you use ImageNet pre-trained weights for medical data?

Anyone who has worked with medical data feels uneasy about using ImageNet pre-trained weights as a starting point because of two reasons –

1. **Medical data looks different than ImageNet data.** For example, in the case of an x-ray image, each pixel represents the density of the material. The x-ray image is, therefore, grayscale. On the other hand, ImageNet consists of natural color images where each pixel measures the reflectance of some surface in the real world.
2. **Medical datasets are small:** Medical datasets are usually small – sometimes a few hundred images or a few thousand if you are lucky. We have to freeze most network layers to prevent overfitting and train only the last few layers. In other words, we are relying on pre-trained weights very heavily even when we know medical data does not look like images in ImageNet.

So, on the surface, it looks like using ImageNet pre-trained weights for medical data may not be a good idea.

5. ImageNet pre-trained weights have produced Human-Level accuracy.

It is time to be surprised!

Before we ditch the idea of using ImageNet pre-trained weights, it makes sense to dig deeper to see what practitioners are doing in the field.

Let's look at a few examples.

1. CheXNet achieved radiologist-level pneumonia detection on chest x-rays in 2017. They used a network based on DenseNet and used ImageNet pre-trained weights.
2. Dermatologist-level classification of skin cancer was achieved by fine-tuning Inception V3 with ImageNet pre-trained weights.
3. Expert level detection of diabetic retinopathy in retinal fundus photographs was achieved by fine-tuning Inception V3 on ImageNet pre-trained weights.
4. Human expert level diagnosis of choroidal neovascularization, diabetic macular edema, and drusen using Optical Coherence Tomography (OCT) images was achieved by fine-tuning Inception V3 with ImageNet pre-trained weights.
5. CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images pretrains a custom model on ImageNet.

Some of these papers are collaborations between top-notch medical teams and leaders in Deep Learning and AI. For example, the CheXNet paper has experts from Harvard Medical School and Dr. Andrew Ng – a leading expert in AI. Similarly, some authors of the skin cancer paper are from the Departments of Dermatology and Pathology, and others are from Electrical Engineering and Computer Science.

These experts in the field make careful choices, and so we can safely conclude that using ImageNet pretrained models is not a bad idea at all. We will see concrete evidence to support this claim in the following section.

In the future, we may have an extensive medical dataset rivaling ImageNet in size, and when that happens, we should probably switch to using weight trained on that large dataset.

6. ImageNet pretrained weights produce better results in chest X-Ray data.

The following paper presents the most comprehensive analysis of transfer learning using popular ImageNet architectures and ImageNet pretrained weights on chest X-ray dataset –

CheXtransfer: Performance and Parameter Efficiency of ImageNet Models for Chest X-Ray Interpretation

They studied 16 different architecture – DenseNet (121, 169, 201), ResNet (18, 34, 50, 101), Inception (V3, V4), MNASNet, EfficientNet (B0, B1, B2, B3), and MobileNet (V2, V3). They used a large chest x-ray dataset called CheXPert for their analysis.

Here is a summary of the conclusions in the paper.

6.1 ImageNet performance does not correlate with CheXpert performance.

Architectures that perform better on ImageNet do not necessarily perform better on CheXNet regardless of whether ImageNet pretrained weights were used or not.

Specifically, "newer architectures generated through search (EfficientNet, MobileNet, MNASNet) underperform older architectures (DenseNet, ResNet) on CheXpert."

In other words, the newer architectures may be overfitting to ImageNet, which probably explains the popularity of older architectures in the medical domain.

I wonder if how these new architectures perform on other transfer learning tasks.

6.2 Choice of Model family > Model Size

The choice of a model family (say DenseNet vs. MobileNet) has a more significant impact on performance than the model's size within the same family (e.g., DenseNet 121 vs. DenseNet169).

6.3 Use ImageNet Pretraining

ImageNet pretraining yields a statistically significant boost in performance across architectures. Smaller architectures benefit a lot more than larger architectures.

It is important to note that a previous paper titled Transfusion: Understanding Transfer Learning for Medical Imaging had concluded that pretraining using ImageNet weights produces negligible performance improvement. They had studied ResNet50 and InceptionV3.

Fortunately, there is no conflict between the two studies. CheXTransfer paper finds that "pretraining does not boost performance for ResNet50, InceptionV3, InceptionV4, and MNASNet but does boost performance for the remaining 12 architectures."

6.4 Truncated Architectures for Best of Both Worlds

The ChexTransfer paper and the Transfusion paper mentioned above conclude that many ImageNet architectures are needlessly large.

An easy way to reduce the model's size while preserving the benefits of pretrained weights is to truncate the final blocks of pretrained models. This way, you can reduce the model's size by 3.5x without affecting the performance.

Truncated architectures are DeepCakes, as in you can have your cake and eat it too!

7. How to solve a medical image classification problem: A Prescription

We have covered a lot in this post. So, let me summarize how to solve a medical image classification problem in easy prescriptive steps.

1. **Data is the King:** It would be best if you planned on spending more than 70% of the time gathering data, getting it annotated from multiple experts, and fixing noise and inconsistencies in the data. We will cover these aspects in a future post, but this video should get you started on data-centric AI problem-solving.
2. **Use a standard ImageNet architecture:** Check the most popular architectures in your domain. For X-Rays, DenseNet121 would be a good choice. We have also had a good experience with Inception V3. Older architectures like AlexNet and VGG are still prevalent in some domains. Before using these ancient architectures, please check if people get good results with slightly newer architectures like DenseNet, Inception, etc. Similarly, I'd stay away from EfficientNet, MobileNet, and MNASNet.
3. **Fine-Tuning / Transfer Learning using ImageNet pre-trained weights:** Always use ImageNet pretrained weights to initialize the model. Depending on the amount of data, fine-tune the model. Often this is accomplished by freezing all but the last one or two layers of the network. If the dataset is small (<1000 samples), you should probably freeze all but the final layer. Use standard hyper-parameters.
4. **Error analysis:** Medical datasets are often small. To understand noise and ambiguity in your dataset, it is crucial to manually check every image where the model is making a mistake. You may find that two experts looking at the same picture differ in their diagnoses. You have to go back and make the data consistent in such cases. Remember, spend most of your time looking at the data.
5. **Optimize model:** Once you have obtained a good baseline model using a standard ImageNet architecture, you may want to experiment with hyperparameter optimization, truncating the model to reduce its size, etc. Remember, this step will give you only half to one percentage point increase in performance. Analyzing and fixing your data may provide you with an order of magnitude higher return on time investment.
6. **Tell your friends about this post:** This step is optional, but it ensures a steady supply of high-quality articles!

8. Feedback

The literature on medical image analysis is vast, and we have simply touched the surface. If I have missed an important point, please feel free to point it out in the comments section. If possible provide a reference paper I can look at. Our goal is to present the best information to the readers.

9. Acknowledgment

We thank Pranav Rajpurkar for valuable pointers.
Feature Visual Credits : Public dataset from Centre for Artificial Intelligence in Medicine & Imaging, Stanford University

Subscribe

If you liked this article, please subscribe to our newsletter. You will also receive a free Computer Vision Resource guide. In our newsletter, we share Computer Vision, Machine Learning and AI tutorials written in Python and C++ using OpenCV, Dlib, Keras, Tensorflow, CoreML, and Caffe.