Transfusion: Understanding Transfer Learning with Applications to Medical Imaging

Maithra Raghu [*1,2] Chiyuan Zhang[*2] Jon Kleinberg[†1] Samy Bengio[†2]

[1]Cornell University
[2]Google Brain

Abstract

With the increasingly varied applications of deep learning, transfer learning has emerged as a critically important technique. However, the central question of how much feature reuse in transfer is the source of benefit remains unanswered. In this paper, we present an in-depth analysis of the effects of transfer, focusing on medical imaging, which is a particularly intriguing setting. Here, transfer learning is extremely popular, but data differences between pretraining and finetuing are considerable, reiterating the question of what is transferred. With experiments on two large scale medical imaging datasets, and CIFAR-10, we find transfer has almost negligible effects on performance, but significantly helps convergence speed. However, in all of these settings, convergence without transfer can be sped up dramatically by using only mean and variance statistics of the pretrained weights. Visualizing the lower layer filters shows that models trained from random initialization do *not* learn Gabor filters on medical images. We use CCA (canonical correlation analysis) to study the learned representations of the different models, finding that pretrained models are surprisingly similar to random initialization at higher layers. This similarity is evidenced both through model learning dynamics and a transfusion experiment, which explores the convergence speed using a subset of pretrained weights.

1 Introduction

As the applications of deep learning have diversified, transfer learning has become a critical tool for model development. In computer vision, the canonical implementation for transfer learning is to pretrain the model on a large dataset, such as ImageNet, and then "finetune" the network on the target data.

Despite the popularity of transfer learning, its exact benefits are poorly understood, with recent papers (He et al., 2018; Ngiam et al., 2018; Kornblith et al., 2018) challenging deep-rooted prior assumptions. One such core assumption is that transfer learning is helpful due to the reuse of learned features. Whether this is really the primary factor, above all other ancillary benefits of pretraining (e.g. better weight conditioning), remains a major open question.

This question is especially relevant in the context of medical imaging. On the one hand, transfer learning from ImageNet has been almost universally adopted Shin et al. (2016); Rajpurkar et al. (2017); Wang et al. (2016); Gulshan et al. (2016); Liu et al. (2017); Raghu et al. (2018), as models must learn to process large

[*]Equal Contribution
[†]Equal Contribution

medical images with much less data. But on the other hand, differences between (natural) images in ImageNet and medical images are considerable (Figure 1).

In this paper, we perform a detailed study of the effects of transfer learning, with a particular focus on medical imaging. Our main contributions are the following:

(1) On two different large scale medical imaging applications and a transfer task on CIFAR-10, we show that transfer learning typically gives negligible final performance boosts, but results in faster convergence.

(2) Addressing the question of feature reuse vs other ancillary benefits, we show that a significant portion of the convergence speedup can be gained by a better scaling. We propose a new initialization, Mean Var Init, that uses *only* the mean and variance statistics of the pretrained weights to rescale random initialization. Mean Var Init converges much faster than random init, and as fast as sampling from the full empirical pretrained weight distribution.

(3) We analyze the learned features and representations of the different initializations using visualizations and CCA (canonical correlation analysis), finding that lower layers trained from scratch on medical images do *not* learn Gabor filters. This forms a striking contrast with conventional wisdom on low-level image representations, which had posited Gabor filters to be useful across essentially all image families; indeed, even the human visual system contains analogues of Gabor filters.

(4) We identify surprising similarities between pretrained and untrained initializations at higher layers, and validate these results through studying convergence speeds of a *(weight) transfusion* experiment – using a partial subset of pretrained weights.

2 Background and Related Work

In this paper, we study the effects of transfer learning. Transfer learning (in the context of deep neural networks) denotes the process of first training a model on a dataset D, and then training the same model on a target dataset D_{target} (known as finetuing). The crucial point is that when training (finetuning) on D_{target}, the model is initialized with the weights learned on D. These weights are called the *pretrained* weights.

Insights on the most effective methodology and precise effects of transfer learning continue to develop rapidly. In recent work, Ngiam et al. (2018) studies how the choice of pre-training data impacts performance and found it is not a simple 'the more the better' relation. Kornblith et al. (2018) studies the effectiveness of ImageNet pretraining on a number of tasks, and found that while ImageNet architectures generalize well across datasets, ImageNet features are less general than previously suggested. Similarly, He et al. (2018) found that for object detection and instance segmentation, training from random initialization achieves comparable performances to ImageNet pretraining. Results from Geirhos et al. (2019) shows that ImageNet trained ConvNets are strongly biased towards texture features. Earlier work has also investigated transfer learning, such as Yosinski et al. (2014), comparing fine-tuning and frozen features effects on transfer. Prior work on understanding transfer learning is mostly based on some notion of similarity between the source and target tasks (not the case here) with both empirical (Ge and Yu, 2017; Cui et al., 2018; Zamir et al., 2018) and theoretical analysis (Blitzer et al., 2008; Ben-David et al., 2010; Galanti et al., 2016). Less work exists comparing learning dynamics and features between transfer learning and learning from scratch, our main focus.

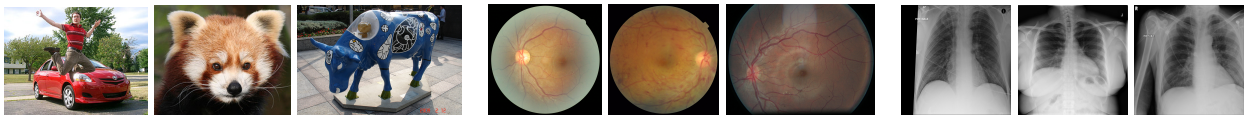


Figure 1: Example images from the *Imagenet*, the *fundus photographs*, and the *ChestXray14* datasets, respectively.

3 Data and Experimental Setup

Our primary dataset consists of *fundus photographs* Gulshan et al. (2016), large 587×587 images of the back of the eye. These images can be used to diagnose the subject with a variety of different eye diseases. One such eye disease is Diabetic Retinopathy (DR), which remains a leading cause of blindness despite being treatable if caught early enough Ahsan (2015). DR is graded on a five class scale, from no DR (grade 1) to proliferative DR (grade 5) AAO (2002). There is an important threshold at grade 3 (moderate DR), with grades 3 and above constituting *referable DR* (requiring immediate specialist attention) and grades $1, 2$ corresponding to *non-referable DR*. Similar to prior work Gulshan et al. (2016), we train a deep neural network (a Resnet-50, He et al. (2016)) to classify each image as one of the 5 DR grades, and evaluate the model by computing the area under the sensitivity-specificity curve (AUC) for identifying referable DR.

We also study a second medical imaging dataset, *ChestXray14*, Wang et al. (2017) which consists of frontal X-ray images (resized to 224×224), which can be used to diagnose 14 different thoracic pathologies, including pneumonia, atelectasis, hernias and others. Using the setup of Rajpurkar et al. (2017) we train a DenseNet-121 on this dataset, evaulating with AUC. [1]

Figure 1 shows some example images from both datasets, as well as the Imagenet dataset, from which the transfer learning is performed. The example images demonstrate drastic differences in visual features among those datasets.

Finally, we conduct these experiments on natural images, studying transfer learning from CIFAR-100 to CIFAR-10 (Krizhevsky and Hinton, 2009), using Resnet-50. Surprisingly, we find that some of the same conclusions also hold with natural image data, demonstrating the potential of these findings to be more broadly applicable.

We use a fixed set of hyperparameters for all experiments on a dataset. Further implementation details can be foound in the Appendix.

4 The Effects of Transfer

In this section, we study the effect of transfer learning on the fundus photograph data. We find that transfer does not help significantly with performance, but does enable faster convergence. To understand the role of feature reuse in transfer, we define the *Mean Var Initialization*, which uses no pretrained features but only the same scaling (via mean and variance statistics) of the pretrained weights. Training from this init both matches transfer performance and significantly speeds up random init convergence, even as training dataset size is decreased. We perform additional experiments comparing to sampling from the full empirical distribution and the effect of batch norm layers.

[1] We build our experiments based on the open source implementation at `https://github.com/zoogzog/chexnet`.

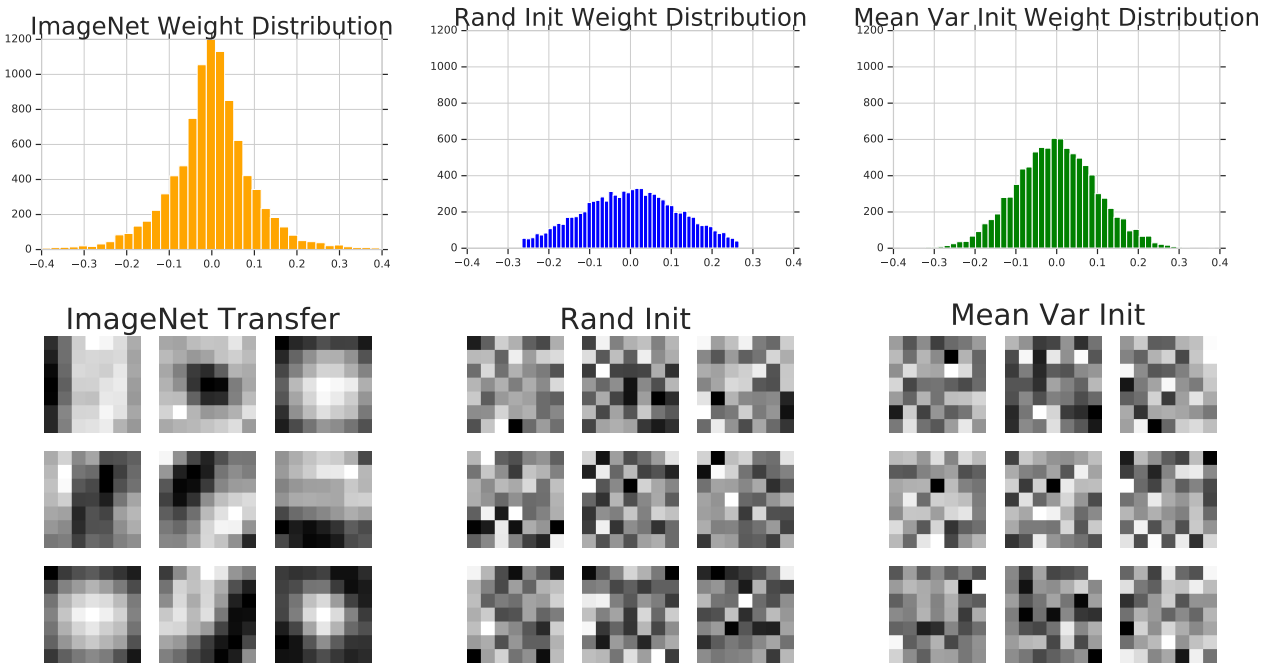


Figure 2: **Distribution and filter visualization of weights initialized according to pretrained ImageNet weights, Random Init, and Mean Var Init**. The top row is a histogram of the weight values of the the first layer of the network (Conv 1) when initialized with these three different schemes. The bottom row shows some of the filters corresponding to the different initializations. Only the ImageNet Init filters have pretrained (Gabor-like) structure, as Rand Init and Mean Var weights are iid. In Section 6, we further visualize and analyze these filters before and after training.

4.1 The Mean Var Initialization

Neural network weights are typically initialized as iid draws from some distribution. A very popular initialization scheme is the Xavier initialization from Glorot and Bengio (2010), where a weight w in layer l is drawn from a Gaussian. Letting n_{l-1} be the number of neurons in layer $l-1$ and n_l the same for layer l, the Gaussian distribution for the weights is $\mathcal{N}(0, \frac{2}{n_{l-1}+n_l})$. This scaling is applied to prevent exploding or vanishing gradients, and the factor of 2 to work with the ReLU activation.

The Mean Var Init is very similar to random initialization but transfers scale information from the pretrained weights. It is a simple way to test the benefits arising solely from the weight scaling found by transfer learning. More specifically, let W' be our pretrained weights, and $W'^{(l)}$ the weights at layer l. We initialize our new weights w at layer l by drawing from $\mathcal{N}(\mu'^{(l)}, (\sigma'^{(l)})^2)$, where $\mu'^{(l)}, (\sigma'^{(l)})^2$ are the mean and variance of $W'^{(l)}$. In Figure 2, we show the distribution of the first layer weights under the three different initializations, and also visualize some of the corresponding filters. Only the ImageNet filters have clear pretrained features.

4.2 Effects on Performance and Convergence

As described in Section 3, we train a Resnet-50 (He et al., 2016) on the fundus photograph data. We evaluate the performance of: (1) transfer learning from ImageNet (2) training from random initialization on the

Training Method	AUC
ImageNet Transfer	96.6 ± 0.041%
Random Init	96.4 ± 0.064%
Mean Var Init	96.5 ± 0.026%

Table 1: **Performance AUC (the higher the better) of transfer from ImageNet, Random Init, Mean Var Init on the fundus dataset are all comparable.** All three initialization schemes give similar performance (averaged over 5 runs).

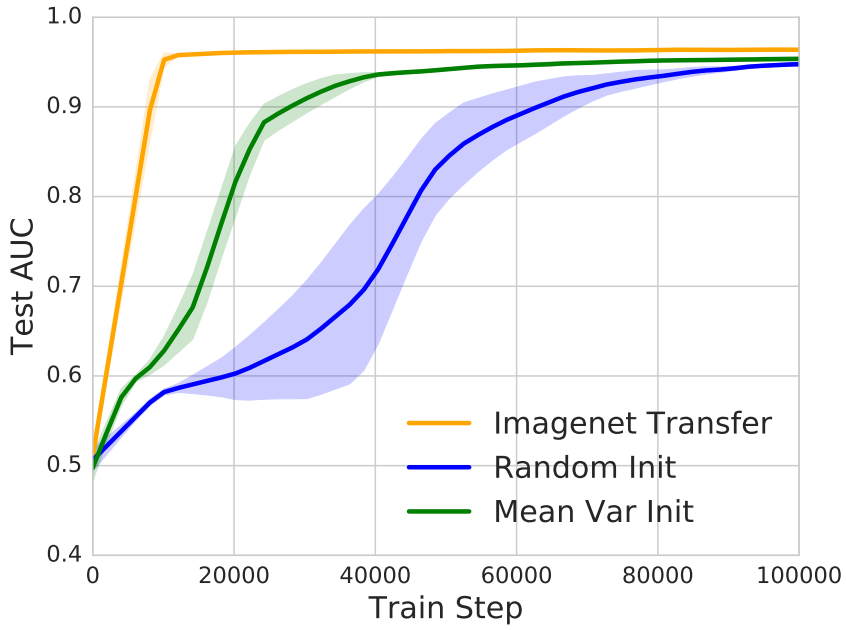


Figure 3: **Convergence of ImageNet init, Random Init and Mean Var over the first 100000 training steps.** We see that the ImageNet init converges the fastest, but the Mean Var init provides a significant speedup over Random Initialization. Each line is averaged over 5 runs. As the Mean Var init destroys the structure of the ImageNet filters (Figure 2), this demonstrates that some of the speedups of transfer can be attributed to better scaling.

fundus dataset (3) training from the Mean Var init. The AUC of these three methods, averaged over 5 runs, is shown in Table 1. We observe that all three training methods (transfer, random init, mean var init) achieve comparable performance, with only very small gains for transfer learning.

However, the three methods have noticeably different convergence speeds. Figure 3 shows the convergence of the three different models over the first 100,000 training steps, with the plot to full convergence (Figure 11) in the Appendix. Transfer learning from ImageNet converges the fastest, much faster than random initialization. But we also see that the Mean Var init converges significantly faster than random initialization, despite the fact it doesn't reuse pretrained features. (Figure 2). This suggests that some of the speedup with transfer from ImageNet is due to better scaling of the weights, instead of solely the learned features.

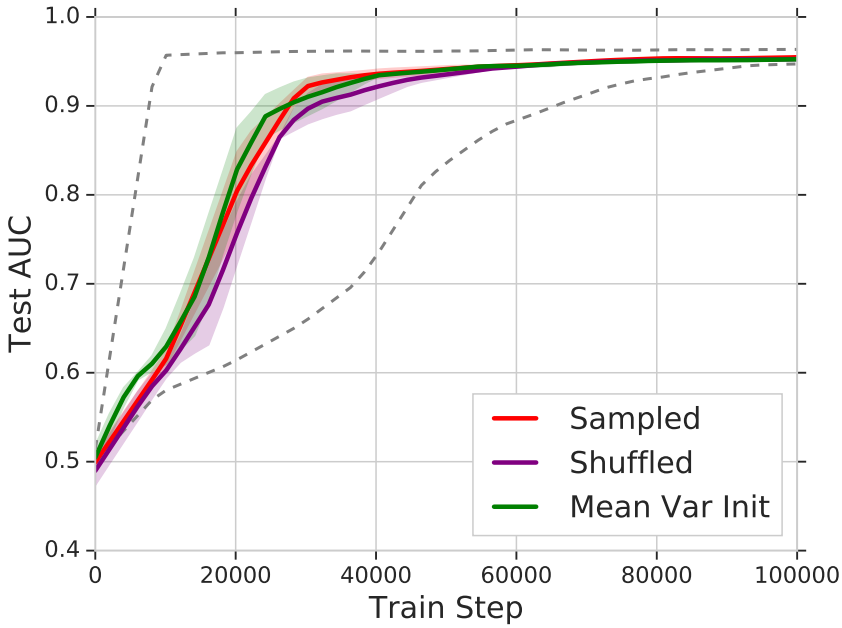


Figure 4: **The Mean Var Init converges with a similar speed to using the full empirical distribution of the pretrained ImageNet weights.** The plots show the convergence speed of initializing by sampling from the empirical ImageNet weight distribution, and initializing by randomly shuffling the pretrained weights (i.e. sampling without replacement). We see that Mean Var converges at a similar speed to using the full empirical distribution. All lines are averaged over 3 runs, and the dashed lines show the convergence of the Imagenet init and the Random init as a reference.

4.3 Mean Var Init vs Using Knowledge of the Full Empirical ImageNet Weight Distribution

In Figure 2, we see that while the Mean Var Init might have the same mean and variance as the ImageNet weight distribution, the two distributions themselves are quite different from each other. We examined the convergence speed of initializing with the Mean Var Init vs initializing using knowledge of the entire empirical distribution of the ImageNet weights.

In particular, we looked at (1) *Sampling Init:* each weight is drawn iid from the full empirical distribution of ImageNet weights (2) *Shuffled Init:* random shuffle of the pretrained ImageNet weights to form a new initialization. (Note this is exactly sampling from the empirical distribution without replacement.) The results are illustrated in Figure 4. Interestingly, Mean Var is very similar in convergence speed to both of these alternatives. This would suggest that further improvements in convergence speed might have to come from also modelling correlations between weights.

4.4 Batch Normalization Layers

Batch normalization layers Ioffe and Szegedy (2015) are an essential building block for most modern network architectures with visual inputs. However, these layers have a slightly different structure that requires more careful consideration when performing the Mean Var init. Letting x be a batch of activations,

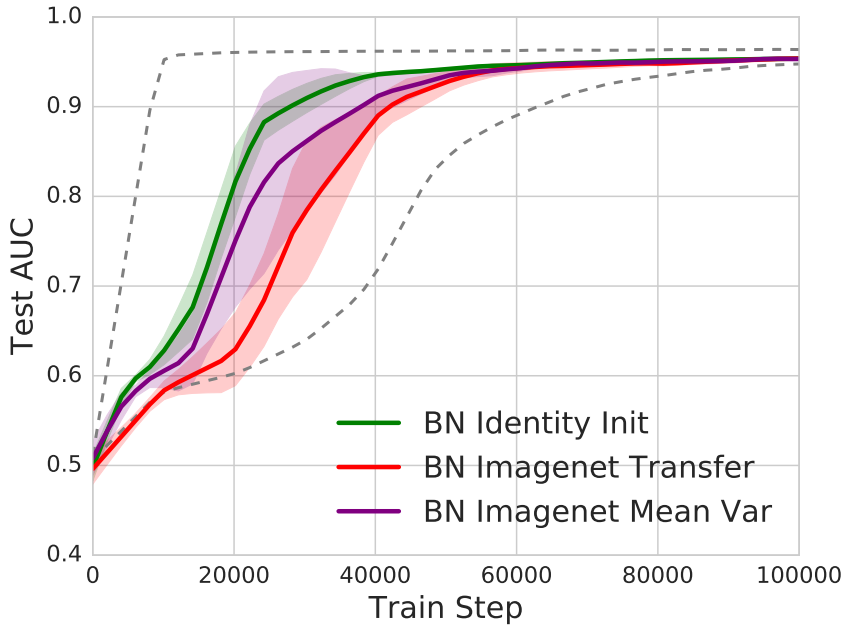


Figure 5: **Comparing different ways of importing the weights and statistics for batch normalization layers.** The rest of the layers are initialized according to the Mean Var scheme. The two dashed lines show the convergence of the Imagenet init and the Random init for references. The lines are averaged over 5 runs.

Training Method	50k	100k	150k	200k	300k
ImageNet Transfer	95.9 ± 0.1%	96.1 ± 0.1%	96.2 ± 0.02%	96.3 ± 0.1%	96.6 ± 0.041%
Random Init	95.5 ± 0.15%	96.0 ± 0.06%	96.0 ± 0.1%	96.0 ± 0.1%	96.4 ± 0.064%
Mean Var Init	95.6 ± 0.08%	96.0 ± 0.02%	96.1 ± 0.08%	96.1 ± 0.05%	96.5 ± 0.026%

Table 2: **Performance AUC (the higher the better) of transfer from ImageNet, Random Init, Mean Var Init as train dataset size is varied.** There is a slightly larger gap between transfer learning and Random/Mean Var init for very small amounts of train data (50k), but otherwise results remain comparable. All results are averaged over 3 runs.

batch norm computes

$$\gamma\left(\frac{(x-\mu_B)}{\sigma_B+\epsilon}\right)+\beta$$

Here, γ, β are learnable scale, shift parameters, and μ_B, σ_B are an accumulated running mean and variance over the train dataset. Thus, in transfer learning, μ_B, σ_B start off as the mean/variance of the ImageNet data activations, unlikely to match the medical image statistics. Therefore, for the Mean Var Init, we initialized all of the batch norm parameters to the identity: $\gamma, \sigma_B = 1, \beta, \mu_B = 0$. We call this the *BN Identity Init*. Two alternatives are *BN ImageNet Mean Var*, resampling the values of all batch norm parameters according to the ImageNet means and variances, and *BN Imagenet Transfer*, copying over the batch norm parameters from ImageNet. We compare these three methods in Figure 5, with non-batchnorm layers initialized according to the Mean Var Init. Broadly, they perform similarly, with *BN Identity Init* (what we use) performing slightly better. We observe that *BN ImageNet Transfer*, where the ImageNet batchnorm parameters are transferred directly to the medical images, performs the worst.

Disease	ImageNet Init	Mean Var	Rand Init
Pneumonia	0.762 ± 0.006	0.765 ± 0.003	0.751 ± 0.005
Cardiomegaly	0.901 ± 0.004	0.910 ± 0.002	0.908 ± 0.003

Table 3: **AUCs for diseases on ChestXray14**, including the primary task of pneumonia detection, and one other thoracic disease. We observe that Mean Var performs slightly better than ImageNet init on pneumonia detection. But overall results vary slightly disease by disease. Full results are in the Appendix.

4.5 Varying the Amount of Training Data

A popular use case for transfer learning is when the target dataset, D_{target} is very small. Here, transfer learning has the potential to both prevent overfitting and help with learning better features through the pretraining. We compared transfer to random init/mean var when varying the size of D_{target}.

The performance results are shown in Table 2. For very small amounts of data (50k examples), there is a slightly larger gap between the ImageNet init and the Random/Mean Var inits, but otherwise the performance remains comparable. Figure 13 in the Appendix plots the corresponding convergence speeds. Random init shows some convergence variance over dataset size, while Mean Var and ImageNet are both quite stable and similar to the full dataset.

5 Other Datasets

We repeated our core experiments on two additional datasets. The first is another large scale medical imaging dataset, *ChestXray14* Wang et al. (2017), where again transfer learning is performed from ImageNet. The other is on natural images, on CIFAR-10 Krizhevsky and Hinton (2009). Here the transfer is from CIFAR-100.

5.1 Other Datasets: ChestXray14

Chest X-rays are a common diagnostic test that can be used by radiologists to diagnose different thoracic diseases. In particular, chest x-rays can be used to diagnose pneumonia, which is an especially common illness. In Rajpurkar et al. (2017), a deep neural network is used to diagnose pneumonia directly from the chest X-ray images. The neural network model is a DenseNet121 Huang et al. (2017), which is first trained on ImageNet, and then finetuned on the x-rays. As in Section 4, we study the effect of the three different methods (transfer from ImageNet, Random Init and Mean Var) on model performance and convergence in this setting.

A subset of our performance and convergence results are shown in Table 3 and Figure 6, with the full results in the Appendix. (Note the ChestXray14 dataset has 13 other disease labels beside pneumonia.) Performance wise, we see that for most diseases, including the primary task of pneumonia detection, all three methods perform comparably, with Mean Var sometimes outperforming pretraining on ImageNet. However, there are a couple of diseases (e.g. Hernia) for which ImageNet pretraining performs better. Convergence wise, we again see that ImageNet converges the fastest, followed by Mean Var and then Random init, with different diseases having different convergence patterns.

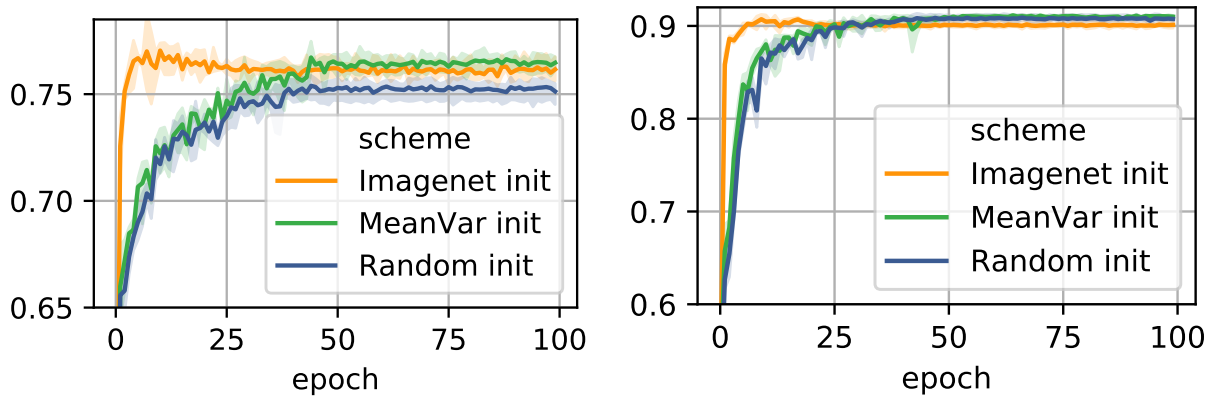


Figure 6: **Convergence results for different diseases from ChestXray14 dataset.** The left plot shows Pneumonia and the right Cardiomegaly. ImageNet converges the fastest, followed by Mean Var and then Random Init, though there is some disease-dependent variability. Full results in the Appendix.

Init	10% data	20% data	40% data	60% data	80% data	100% data
CIFAR100	0.826 ± 0.002	0.885 ± 0.002	0.918 ± 0.002	0.934 ± 0.002	0.943 ± 0.002	0.949 ± 0.002
Mean Var	0.774 ± 0.011	0.863 ± 0.002	0.914 ± 0.001	0.931 ± 0.002	0.941 ± 0.002	0.948 ± 0.002
Random	0.733 ± 0.013	0.846 ± 0.014	0.916 ± 0.004	0.936 ± 0.004	0.946 ± 0.001	0.953 ± 0.002

Table 4: **Test accuracy on CIFAR10 with the different initialization schemes and varying train data size.** Despite the similarites between CIFAR-100 and CIFAR-10, transfer learning does not dominate the other inits except in the very small data setting.

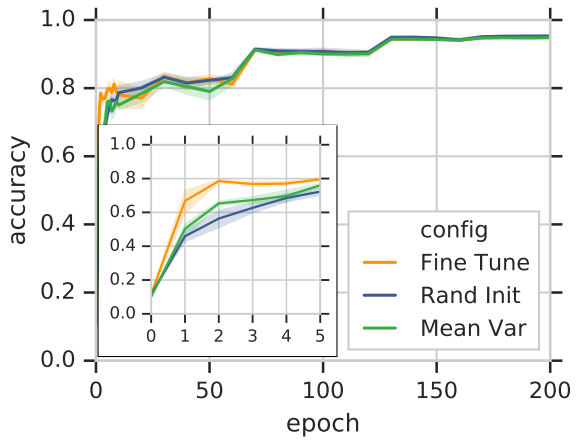
5.2 Other Datasets: CIFAR-100 to CIFAR-10

We also run similar experiments on the CIFAR-10 dataset Krizhevsky and Hinton (2009), with pre-trained weights from the CIFAR-100 dataset. Unlike our medical images and Imagenet data that are different in the underlying visual features and statistics, the images of both the CIFAR datasets are from the *80 million tiny images* dataset (Torralba et al., 2008), therefore are more closely related. As described in Section 3, we compare pretraining a Resnet50 on CIFAR-100 and finetuning on CIFAR-10 to using the Random/Mean Var Inits and across different training dataset sizes.

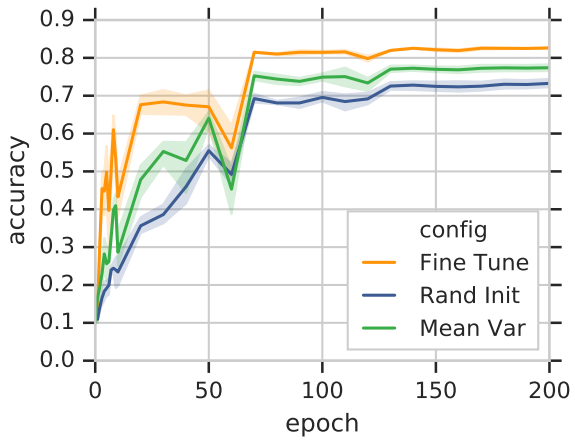
As the images from CIFAR-100, CIFAR-10 are so closely related, we might expect transfer learning to outperform random init/mean var. However, the performances (Table 4) remain comparable except for when there is very little data. Figure 7 right shows convergence in this small data regime, where there is a clear ordering in convergence speeds between the different methods, supporting earlier results. With more data, convergence is approximately the same for all inits (likely due to this being an easier task.)

6 Analysis of Filters and Representations

So far, we have studied performance and convergence properties of 1) transfer learning 2) random init 3) mean var init. A different line of exploration, directly related to the overarching question, is determining whether these three methods learn similar features and representations. To the contrary, we find that particularly in the lower layers, they differ significantly in their learned representations. In fact, both



(a) Full CIFAR-10 training set (with zoom in for the first 5 epochs).



(b) Subset of 10% CIFAR-10 training set.

Figure 7: **Learning curves (of the test accuracy) on CIFAR-10 by finetuning from CIFAR-100.**

Random Init and the Mean Var Init do *not* learn Gabor filters at the lowest layer when trained on medical images.

In view of these surprising contrasts in basic properties, we undertake an extensive evaluation to compare both untrained and trained representations under our different approaches, using both visualization of filters and activations as well as CCA (Canonical Correlation Analysis). In comparing the middle and higher layers, we find clear differences between untrained and trained representations, as well as some surprising similarities between the different untrained initialization schemes. Roughly speaking, the strong differences between the different initializations at lower layers begin to go away as we move to higher layers, and all the untrained initializations start to resemble each other in their basic properties more than they resemble their trained states.

6.1 Methods

To perform this analysis, we use two main methods:

Filter and Activation Visualization We visualize both the filters (the convolutional kernel for a channel) and hidden activations (the output of a channel for a single datapoint) with a heatmap.

CCA (Canonical Correlation Analysis) We also compare the hidden representations of different neural networks. This is challenging to do naively, because there is no alignment between the hidden layers of two networks: a single neuron in one net might correspond to a group of neurons in different net, orderings might be permuted, etc.

However, recent work Raghu et al. (2017); Saphra and Lopez (2018); Morcos et al. (2018); Magill et al. (2018) has shown the effectiveness of CCA in performing representational comparisons across neural networks. CCA measures the similarity between the representations learned by two layers L_1, L_2. L_1 and L_2 can be same/different layers in the same network or same/different layers in different networks. The activations of L_1, L_2 over a set of datapoints are computed, and CCA is then used to align these representations. Due to the invariance of CCA to linear transformations, this alignment can be done in a meaningful way, accounting for permutations, neurons mapping to groups of neurons, etc.

Figure 8: **The filters learned from Random Init/Mean Var are not Gabor-like.** We visualize the filters of Conv 1 of the three inits on medical data, the top row showing fundus images and the bottom ChestXray-14. We show filters before training, after training and the difference between the two. The far left images shows a classic example of Gabor filters, at initialization with the ImageNet weights. These features are not at all present in either the learned filters or the difference for the Rand Init/Mean Var models.

The end result is a scalar similarity score in $[0, 1]$ with 0 meaning the representations of L_1, L_2 are not at all similar, and 1 that they are identical. We refer the reader to Morcos et al. (2018) for a detailed overview of the method, and the Appendix for additional details.

6.2 (The Lack of) Gabor Filters

It has been well documented that convolutional neural networks trained on computer vision tasks (usually natural image based) learn Gabor filters Krizhevsky et al. (2012); Yosinski et al. (2014). This has led to the emergence of a conventional wisdom that they are essentially universal across image applications, a view consistent with the fact that the human visual system has Gabor filters as well. Surprisingly, therefore, models trained from scratch on medical images do *not* automatically learn Gabor filters.

In Figure 8, we visualize some of the filters of the first conv layer of models trained on the fundus dataset from Section 4 and on the ChestXray14 dataset, Section 5.1. We show the filters at initialization, after training, and the *difference*, which illustrates how the filters changed during training. For both Random Init and Mean Var Init, we see that neither the trained filters or the difference is Gabor-like. By contrast, a model trained on ImageNet from random initialization quickly learns Gabors (see Figure 14 in Appendix E).

These observations suggest that Gabor filters may not be universally useful (or at least universally learned) across all image types, with medical images providing one important example. They also reiterate the question of how much the speedups of transfer learning are due to the necessity of transfer-learned features versus other ancillary factors.

6.3 CCA Similarity of Hidden Representations

For higher layers in the network, instead of visualizing the many filters, we apply CCA to understand their representational properties. Building off of Figure 2, we use CCA to compare the similarity of the representations before and after training for different layers and across the three different inits. The top of Figure 9 shows heatmaps of these cross comparisons at the end of Block 1, Block 2, Block 3, Block 4 of Resnet-50 He et al. (2016). Each block consists of four 'units' which themselves consist of three conv-batchnorm-relu progressions as well as a skip connection.

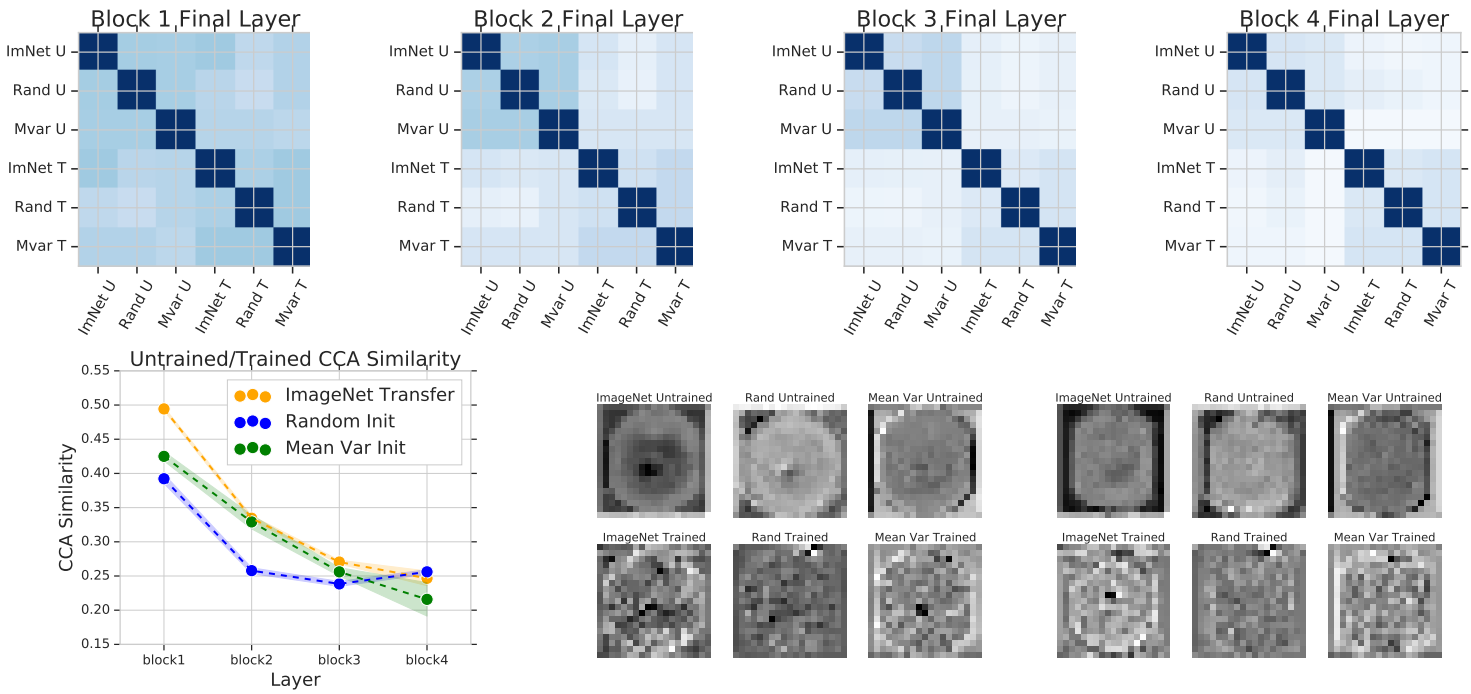


Figure 9: **Top row: heatmaps of CCA similarity of untrained/trained representations at different layers. Bottom left: line plot of untrained/trained CCA similarity. Bottom right: raw activations in Block 3 for two fundus images.** The top row of heatmaps visualize all pairwise comparisons between untrained/trained representations of the three inits for different layers. With this and the bottom left plot, we see that up to Block 2, pretrained/finetuned ImageNet are more similar than the analogous values for Random Init/Mean Var. At Block 3, 4 the heatmaps show all three inits are pairwise similar, matched by visualizing Block 3 activations, bottom right.

Because each block's final layer has a different number of neurons and channels, the CCA similarities for each of the heatmaps is at a slightly different scale Raghu et al. (2017). Nevertheless, we can observe some interesting patterns. Firstly, we note that in lower layers (Block 1 final, Block 2 final), the ImageNet untrained representations are closer to their trained versions than for the Mean Var init and Random init. We plot this directly in the bottom left pane of Figure 9. By Block 3, Block 4, we see that all models are equally (dis)similar to their trained representations.

Another pattern we see is the emergence (in Block 3, Block 4) of two 3x3 blocks of comparatively greater similarity, one on the top left and the other on the bottom right. The block on the bottom right is comparisons of representations of the trained ImageNet/Random/Mean Var models. As all these models solve the same task, similarity between their trained representations in higher layers, is not unexpected.

The similarity of the 3x3 block at the top left however, corresponds to representational similarity of the three untrained models. The similarity of Mean Var untrained to Rand Init untrained is not entirely counter-intuitive, at heart stating that different random convolutional filters have correlated activations. This helpful inductive bias of random convolutional layers has been studied before in the literature Ulyanov et al. (2018); Mahendran and Vedaldi (2015).

But the surprise is the similarity of the ImageNet init representations to Rand Init and Mean Var. We conjecture that due to the significantly different statistics of ImageNet and the fundus dataset, at higher layers, even the ImageNet init acts like random convolutional filters on this dataset. This is further evidenced by the bottom left of Figure 9, where we visualize the activations for two images at the end

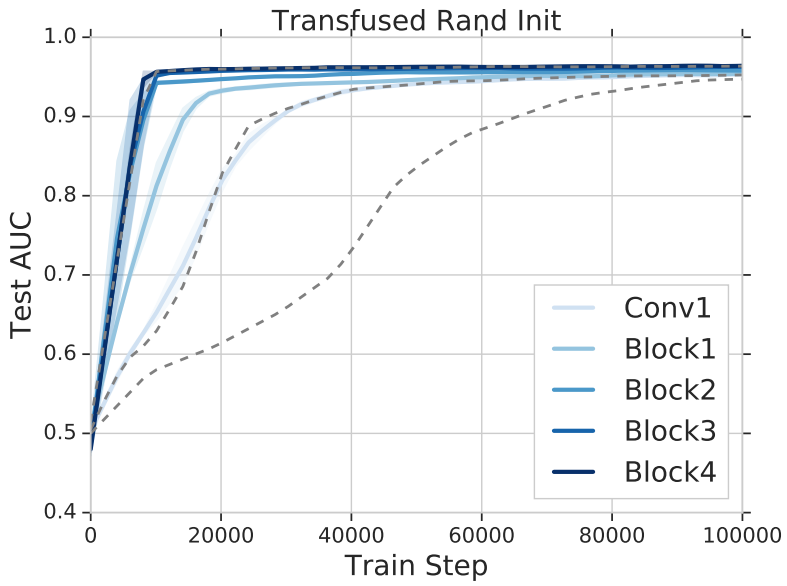


Figure 10: Convergence when we initialize with all weights up to some layer with ImageNet weights, and the rest with random init. As predicted by Section 6.3, ImageNet weights up to Block 2 results in the same convergence speed as ImageNet transfer. Dotted lines showing Imagenet/Meanvar/Rand inits for reference.

of Block 3 of the three models. The top row shows the raw activations for the untrained models, and the bottom row the trained activations. Visually, we can see clear similarities between the untrained activations.

In summary, the observations in Figure 9 indicate that in the lower layers, the untrained ImageNet init is most similar to its trained representation, but at higher layers, all three initializations have functionally very similar properties, and are equally dis(similar) to their trained states. Additional results are in the Appendix.

6.4 Weight Tranfusion

Motivated by Figure 9 (bottom left), we study the effects on convergence when performing a *weight transfusion*: pretrained weights up to a layer L and the rest initialized with Mean Var or Random Init. Figure 9 indicates that transfusing weights up to Block 2 should result in a similar convergence speed to the full ImageNet init, (bottom left plot shows bigger difference in Blocks 1, 2; heatmaps show similarity between all three inits in Block 3, 4). This is indeed what is we observe in Figures 10, 18.

7 Discussion

Even as transfer learning has grown in scope, accounting for its effectiveness has remained an important open problem. The question is particularly compelling for medical images, where transfer learning has enjoyed success despite the clear differences between medical images and natural image domains. Is transfer learning fundamentally drawing its power through feature reuse, or is better conditioning of the weight distribution playing an important role? Our work addresses these questions with a focus on

medical imaging applications. We show that transfer learning gives only minimal final performance gains, but significantly improves convergence speed. In this, *weight scaling* plays a key role: we identify Mean Var initialization as a simple way to use only scaling information from the pretrained weights, but to achieve substantial convergence gains, equal to i.i.d. sampling from the full empirical distribution.

We also compare the representations obtained through ImageNet pretraining, Mean Var initialization, and random initialization. At lower layers, the different approaches yield very different representations, with Random Init and Mean Var *not* learning Gabor filters. At higher layers, representational differences between the different approaches begin to disappear, with pretrained ImageNet filters behaving similarly to random initialization. We explore this further through *weight transfusion*, porting some pretrained weights into a network built from random or Mean Var initialization. In future work, it would be interesting to develop weight distributions leading to further improvements, which (Section 4.3) might also model weight correlations.

8 Acknowledgements

The authors thank Jiquan Ngiam and Geoff Hinton for helpful discussions.

References

AAO (2002). *International Clinical Diabetic Retinopathy Disease Severity Scale Detailed Table.* American Academy of Ophthalmology.

Ahsan, H. (2015). Diabetic retinopathy – biomolecules and multiple pathophysiology. *Diabetes and Metabolic Syndrome: Clincal Research and Review*, 51–54.

Ben-David, S., J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan (2010). A theory of learning from different domains. *Machine learning 79*(1-2), 151–175.

Blitzer, J., K. Crammer, A. Kulesza, F. Pereira, and J. Wortman (2008). Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pp. 129–136.

Cui, Y., Y. Song, C. Sun, A. Howard, and S. Belongie (2018). Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4109–4118.

Galanti, T., L. Wolf, and T. Hazan (2016). A theoretical framework for deep transfer learning. *Information and Inference: A Journal of the IMA 5*(2), 159–209.

Ge, W. and Y. Yu (2017). Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI*, Volume 6.

Geirhos, R., P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel (2019). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*.

Glorot, X. and Y. Bengio (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256.

Gulshan, V., L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. Q. Nelson, J. Mega, and D. Webster (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA 316*(22), 2402–2410.

He, K., R. Girshick, and P. Dollár (2018). Rethinking imagenet pre-training. *arXiv preprint arXiv:1811.08883*.

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Huang, G., Z. Liu, L. Van Der Maaten, and K. Q. Weinberger (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269. IEEE.

Ioffe, S. and C. Szegedy (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Kornblith, S., J. Shlens, and Q. V. Le (2018). Do better imagenet models transfer better? *arXiv preprint arXiv:1805.08974*.

Krizhevsky, A. and G. Hinton (2009). Learning multiple layers of features from tiny images. In *Tech Report*.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105.

Liu, Y., K. Gadepalli, M. Norouzi, G. E. Dahl, T. Kohlberger, A. Boyko, S. Venugopalan, A. Timofeev, P. Q. Nelson, G. S. Corrado, J. D. Hipp, L. Peng, and M. C. Stumpe (2017). Detecting cancer metastases on gigapixel pathology images. *CoRR arXiv:1703.02442*.

Magill, M., F. Qureshi, and H. de Haan (2018). Neural networks trained to solve differential equations learn general representations. In *Advances in Neural Information Processing Systems*, pp. 4075–4085.

Mahendran, A. and A. Vedaldi (2015, June). Understanding deep image representations by inverting them. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Morcos, A. S., M. Raghu, and S. Bengio (2018). Insights on representational similarity in neural networks with canonical correlation. *arXiv preprint arXiv:1806.05759*.

Ngiam, J., D. Peng, V. Vasudevan, S. Kornblith, Q. V. Le, and R. Pang (2018). Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*.

Oakden-Rayner, L. (2017). Exploring the chestxray14 dataset: problems. *Wordpress*.

Raghu, M., K. Blumer, R. Sayres, Z. Obermeyer, R. Kleinberg, S. Mullainathan, and J. M. Kleinberg (2018). Direct uncertainty prediction for medical second opinions. *CoRR arXiv:1807.01771*.

Raghu, M., J. Gilmer, J. Yosinski, and J. Sohl-Dickstein (2017). Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, pp. 6076–6085.

Rajpurkar, P., J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *CoRR arXiv:1711.05225*.

Saphra, N. and A. Lopez (2018). Understanding learning dynamics of language models with svcca. *arXiv preprint arXiv:1811.00225*.

Shin, H.-C., L. Lu, L. Kim, A. Seff, J. Yao, and R. M. Summers (2016). Interleaved text/image deep mining on a large-scale radiology database for automated image interpretation. *Journal of Machine Learning Research 17*(107), 1–31.

Torralba, A., R. Fergus, and W. T. Freeman (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence 30*(11), 1958–1970.

Ulyanov, D., A. Vedaldi, and V. Lempitsky (2018). Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454.

Wang, D., A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck (2016). Deep learning for identifying metastatic breast cancer.

Wang, X., Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers (2017). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3462–3471. IEEE.

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

Zamir, A. R., A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese (2018). Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3712–3722.

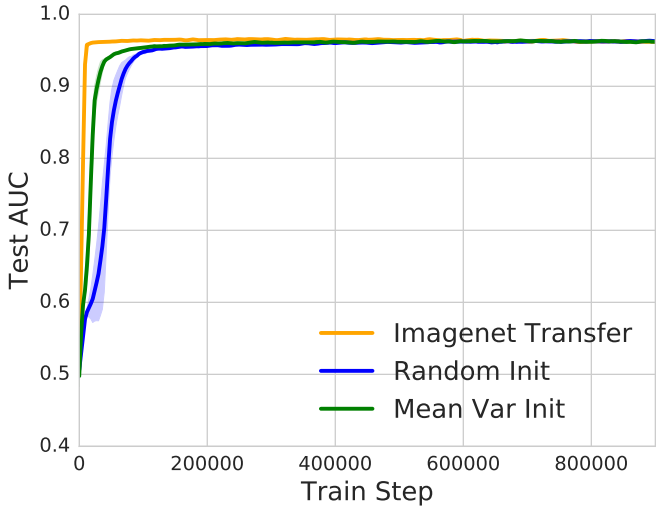


Figure 11: **The full convergence curves of ImageNet init, Random Init and Mean Var.** A zoomed in version of this is shown as Figure 3 in the main text.

A Details on Models and Hyperparameters

For experiments on the fundus photographs, we use a standard Resnet50, replacing the 1000 class ImageNet classification head with a five class head for DR diagnosis. In the experiments with CIFAR-10/CIFAR-100, due to the smaller image sizes, we replace the initial block of Resnet50 from large 7×7 convolution and max pooling with a single 3×3 convolution. The model on the ChestXray14 data is a a DenseNet121 Huang et al. (2017), from the open source implementation at `https://github.com/zoogzog/chexnet`.

The models on the fundus photographs are trained with learning rate 0.001 and a batch size of 8 (for memory considerations.) The Adam optimizer is used. In the ChestXray14 experiment, the models are trained via the Adam optimizer with learning rate 0.0001 and weight decay 10^{-5}. By using multi-GPU data-parallization, we are able to train with a larger batch size (128) than the original implementation. In the CIFAR experiments, both the pre-trained networks on CIFAR-100 and the finetuned models on CIFAR-10 are trained via SGD with momentum 0.9 and weight decay 5×10^{-4}. The learning rate starts at 0.1, and decays by a factor of 0.2 on epoch 60, 120 and 160. The batch size is 128.

B Full Convergence Curves

In the main text, the convergence curves are zoomed in to show the earlier training stage because later in the training process the curves overlap. The full convergence curves are shown here for references. In particular, Figure 11 is the full version for Figure 3 in the main text. Figure 12 corresponds to Figure 5 in the main text.

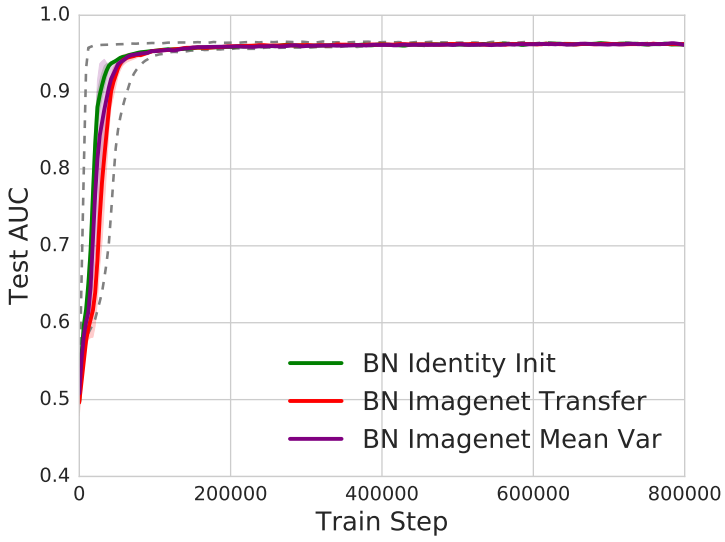


Figure 12: **Comparing different ways of importing the weights and statistics for batch normalization layers.** A zoomed in version of this is shown as Figure 5 in the main text.

C Learning Curves for Varying Data on the Fundus Photographs Dataset

Table 2 in the main text compares the final performance on varying amount of training data for the fundus photograph dataset. In Figure 13, the learning curves are shown to compare the convergences for those cases for reference.

D Other Datasets

D.1 ChestXray14

The ChestXray14 dataset has 14 different disease labels, and we find some variation in the results depending on the disease. For the main task of pneumonia detection, we see that the Mean Var init actually outperforms the ImageNet init, Table 5. There are also small performance gains on a couple of other disease, namely Cardiomegaly and Consolidation.

For a large group of other diseases (Table 5), performance is comparable (within 0.5% or less) across all of the three initialization schemes. However, we do find three diseases, Nodule, Hernia and Emphysema for which the ImageNet init performs significantly better. We suspect that some of these differences might be due to noise in the data labelling process for ChestXray14, which has been well documented Oakden-Rayner (2017).

E Convergence of the First Layer Filters when Training on Imagenet

Figure 14 illustrates the quick learning of Gabor filters from the first layer of a Resnet50 model when trained on ImageNet data.

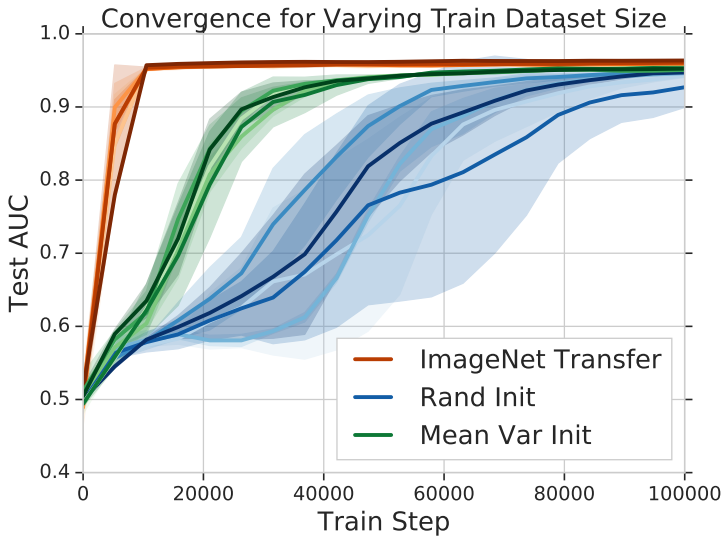


Figure 13: **Convergence of ImageNet init, Random Init and Mean Var as training datast size is varied.** The multiple lines of each color (orange for ImageNet, blue for Rand Init and green for Mean Var) correspond to the convergence of different training dataset sizes averaged over 3 runs. Lighter lines in each color correspond to less data. (See Table 2 for dataset sizes and final performance.) Aside from Random Init showing greater variance in convergence time, the differences between the three methods are relatively stable over dataset size.

F CCA and Learning Dynamics

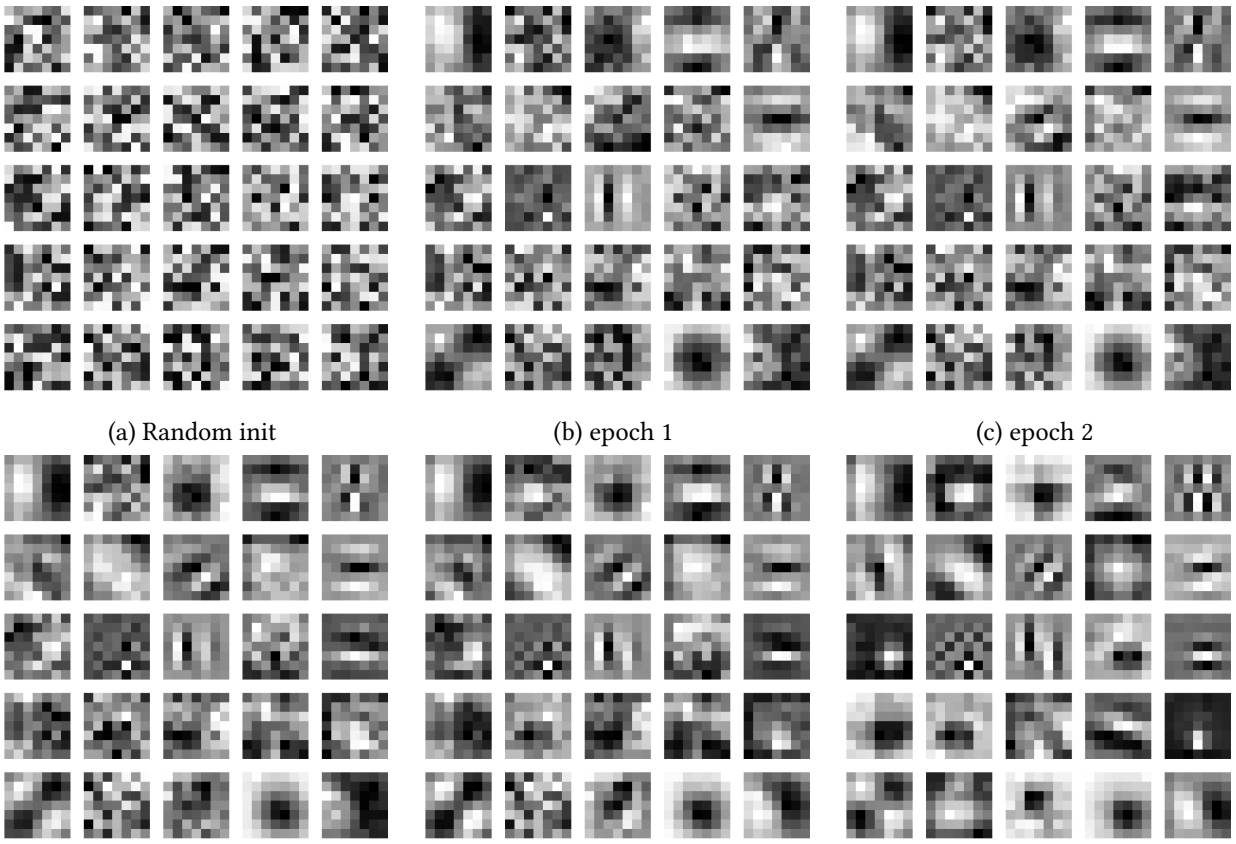
Our implementation of CCA uses the code from https://github.com/google/svcca. To get a CCA similarity score, we compute the mean of all CCA correlation coefficients.

F.1 Learning Dynamics

We further investigate the differences between lower and higher layers by plotting (separately for each initialization) the per layer learning dynamics in Figures 15, 16, 17. We compute CCA similarity for a layer at timestep t with the same layer at full convergence. The results show that layers closer to the input converge much faster than layers higher up the network, as suggested by the heatmap plots of Section 6.3. This pattern remains true across all initializations, and also shows that the higher layers have similar learning dynamics across all three initializations.

F.2 Transfusion with Mean Var

We perform the same experiment as in Figure 10 except using the mean-var initialization for the rest of the weights instead of random initialization. We observe very similar patterns – using pretrained ImageNet weights up to Block 2 gives much faster convergence.



(a) Random init (b) epoch 1 (c) epoch 2

(d) epoch 4 (e) epoch 8 (f) epoch 100

Figure 14: Visualization of the first layer filters from a Resnet50 trained on ImageNet.

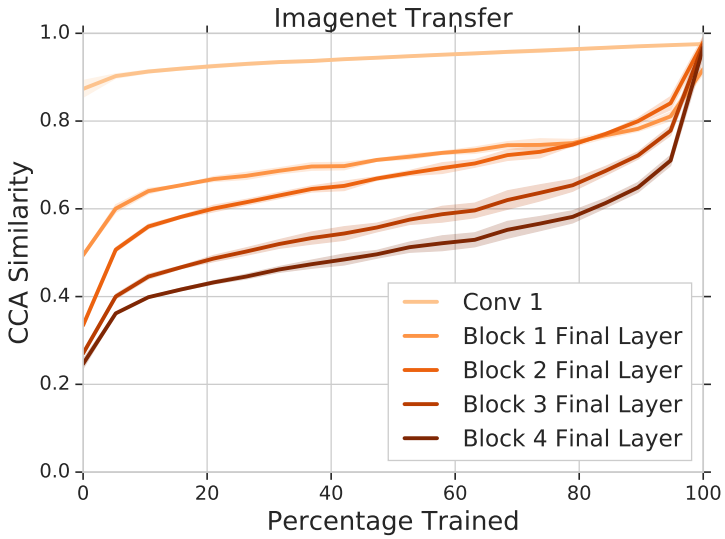


Figure 15: Imagenet learning dynamics

Disease	ImageNet Init	Mean Var	Rand Init
Atelectasis	0.825 ± 0.001	0.811 ± 0.002	0.805 ± 0.001
Cardiomegaly	0.901 ± 0.004	0.910 ± 0.002	0.908 ± 0.003
Effusion	0.882 ± 0.001	0.882 ± 0.001	0.879 ± 0.001
Infiltration	0.706 ± 0.004	0.708 ± 0.001	0.705 ± 0.001
Mass	0.850 ± 0.005	0.854 ± 0.001	0.836 ± 0.009
Nodule	0.793 ± 0.004	0.758 ± 0.007	0.739 ± 0.008
Pneumonia	0.762 ± 0.006	0.765 ± 0.003	0.751 ± 0.005
Pneumothorax	0.881 ± 0.006	0.874 ± 0.001	0.871 ± 0.004
Consolidation	0.806 ± 0.002	0.812 ± 0.003	0.806 ± 0.002
Edema	0.898 ± 0.004	0.894 ± 0.001	0.893 ± 0.001
Emphysema	0.930 ± 0.003	0.908 ± 0.005	0.897 ± 0.007
Fibrosis	0.849 ± 0.006	0.838 ± 0.004	0.833 ± 0.003
Pleural Thickening	0.785 ± 0.002	0.776 ± 0.003	0.767 ± 0.005
Hernia	0.934 ± 0.011	0.920 ± 0.007	0.903 ± 0.006

Table 5: AUCs for all the diseases on ChestXray14.

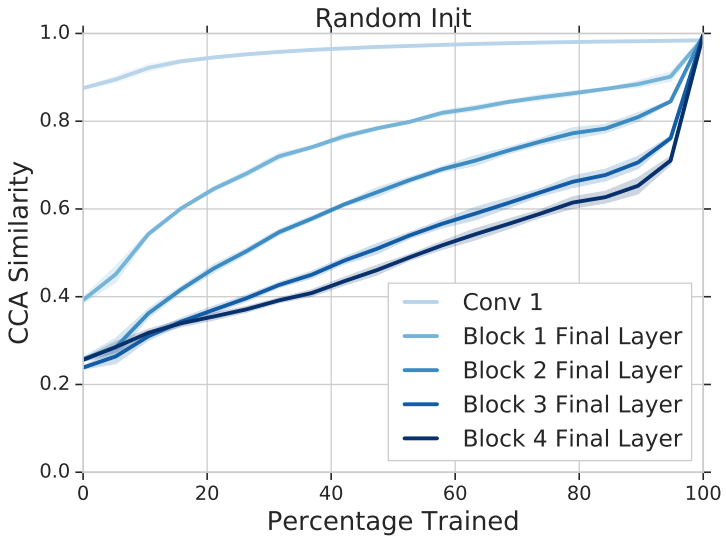


Figure 16: Random init dynamics

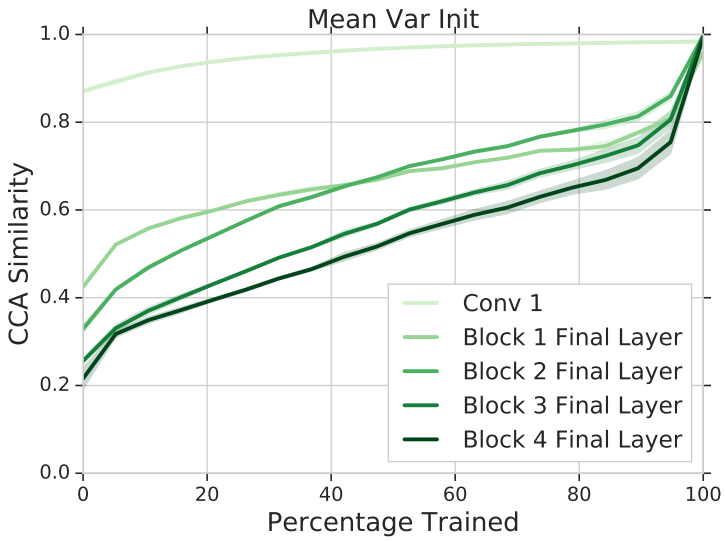


Figure 17: Mean Var learning dynamics

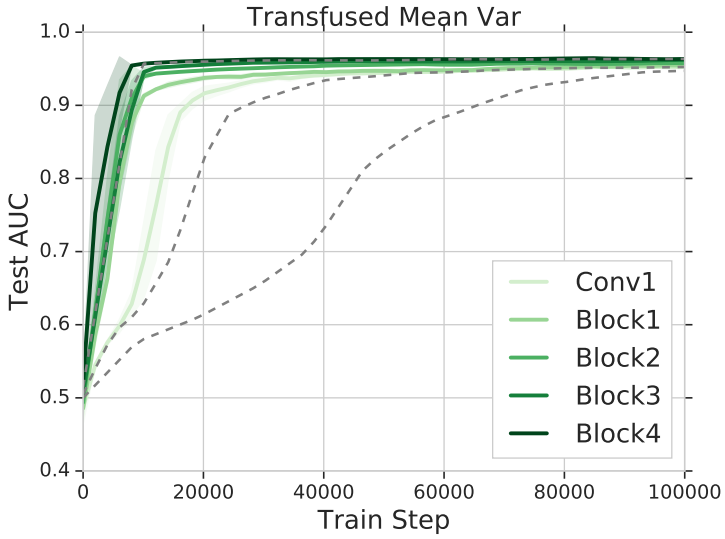


Figure 18: Transfusion with Mean Var Init, similar to Figure 10 in the main text.