Transfusion: Reproducibility Study and Analysis

Pasquale Zingo
Electrical and Computer Engineering
University of Delaware
Newark, DE 19716
patzingo@udel.edu

Austin Brockmeier
Electrical and Computer Engineering
University of Delaware
Newark, DE 19716
ajbrock@udel.edu

Andrew Novocin
Electrical and Computer Engineering
University of Delaware
Newark, DE 19716
andynovo@udel.edu

1 Introduction

In *Transfusion: Understanding Transfer Learning for Medical Imaging* by Raghu et al. (1) (hereafter *"their paper"*), the authors investigate the efficacy of transfer learning from natural image classification to medical image classification. In their paper, a comparison is made between the performance of models that are trained to convergence on ImageNet and then trained on the medical task, and models that are only trained on the task. They found that in all cases the medical task accuracies differed insignificantly between the models with transfer learning and those without, so long as enough data was used. Two state-of-the-art models, ResNet50 and InceptionV3 were compared, as well as a family of smaller CNN models, on the RETINA (2) and CheXpert (3) data sets. We reproduce their work for the state-of-the-art models for the RETINA task on a similar, publicly available dataset, and offer an alternate interpretation of for these experiments.

We suggest that rather than the convergence of random and transfer-initialized models marginalizing the usefulness of transfer techniques, it is interesting that models transfered from such disparate domains do not result in overall worse performance.

2 Background: Transfer Learning

State-of-the-art models used for medical image classification tasks routinely leverage Convolutional Neural Networks (CNNs) trained via transfer learning. Often, these models are pretrained on the ImageNet dataset, which is to say that the weights of the CNN trained on the image classification task are initialized as weights which perform better on classification of ImageNet classes than a random initialization.

Transfer learning is performed as follows. A model is trained to some satisfaction condition on a dataset D_A for some task A, resulting in parameters θ_A. In this work (as in the paper by Raghu et al.), D_A was the ImageNet dataset and task A is image classification. ImageNet contains 1 million images, with each image labeled as member to 1 of 1000 mutually exclusive classes. Next, a model with weights initialized to θ is trained on some different dataset D_B for task B, the desired task, resulting in parameters θ_B. Some goals of transfer learning include:

- Transfer conditioned parameters θ_B outperform parameters θ'_B on task B where the θ'_B is trained from random initialization (no preconditioning from another task).
- Transfer conditioned parameters θ_B achieve minimal test loss faster than θ'_B.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Interpretations of these improvements center on the reuse of low-level features (4) from task A, which can capture a range of basic computer vision basics such as edge and shape detection. Because of differences between task A and task B, there is sometimes a need to modify the structure of the model between tasks. This is solved by removing the last fully connected layer and replacing it with a fully connected layer of appropriate dimension to the output size. While this throws out some expertise from the pretrained parameterization, it is considered to be of little significance since the final layer was the most task specific to task A, and so likely would play little role in learning another task.

In this work, as in their paper, we consider all weights θ_B for uniform update on the target task, both transferred and randomly initialized weights. Other transfer update regimes include:

- Only updating the last layer or layers (if more than one fully connected layer is present at the top of the model). This regime avoids learning any basic vision expertise in the convolutional layers. This task can be modeled as a feed forward network problem where the convolutional layers of the model from task A act as a feature embedding of the low-level features described above.
- Updating the weights of the whole model with non-uniform learning rate, where the learning rate at the bottom (near the input layer) of the model is lower than the learning rate at the top, low-level features at the bottom of the model while being more flexible than the previous method. In contrast, the embedding here can be updated according to the need of the problem, but not nearly as much as in our work.

3 Models

Two models are considered in this work: ResNet50 (5) and InceptionV3 (6). Each of these are deep computer vision models which perform near the state-of-the-art for tasks like ImageNet ILSVRC (ImageNet Large Scale Visual Recognition Challenge) (7).

- ResNet is a CNN which leverages layer-skip-connections and a restricted set of internal function approximations (residual functions) to improve performance, allowing it to be deeper than standard CNN models due to loss of gradient information at equivalent numbers of layers.
- InceptionV3 is a network designed by Google. It makes use of "Inception Blocks" which concatenate the results of multiple convolutional filters of varying size as a single operation, *capturing different resolutions of detail at once.*

4 Data set and Tasks

The RETINA dataset is a collection of retina photographs, each labeled with the grade of Diabetic Retinopathy (DR) that the image was diagnosed with by an expert. The grades are 1 to 5 in increasing order of severity, with 3 and above considered referable DR. While in the original paper a private dataset was used (3), which contains roughly 320k images, this dataset is inaccessible and so we used a smaller public dataset of the same kind, from a Kaggle competition in 2015 (8) which contains just over 35k results. However, the Raghu paper reported results for 'The Very Small Data Regime', which examined models which are only trained on 5000 images from the medical task. This was meant to simulate the impact of transfer learning for tasks with very small data sets. Unlike in the larger data regime, a significant positive difference in the performance of the transfer-initialized model than in the randomly initialized model. Our results will be in the order of magnitude between their results, and we will interpret accordingly. As in the Raghu paper, the images were all resized to 587x587 pixels.

5 Implementation and Methodology

All models were implemented with the pytorch-lightning framework. Both models were instantiated using the torchvision library, which contains classes and pretrained weights for several models, among them those that are considered in this paper. The performance listed in the documentation for

top-5 ImageNet performance is in line with the performance listed in Raghu paper[1]. We start with models pretrained on ImageNet (1000 classes), but work on problems with 5 classes. To address this, we replaced the final layer with a layer of dimension 5, with the parameters leading to it from the penultimate layer (global average normalization of dimension 2048) in line with the section on transfer learning methodology described above. Both models were treated under a multi-class framework where each class corresponds to a level of DR was calculated as a probability of class presence. We performed a 9:1 train-test split and trained models on the target task for 50 epochs. We did not find an explicit stopping criteria mentioned in their paper. For each experiment, 3 models were trained from the same transfer initialization, and conclusions are drawn from statistics about those samples. As in their paper, we used the Adam optimizer with a learning rate of 0.001 for both data sets.

6 Performance Metrics

As in the original paper, for each trained model we calculate the *Receiver Operating Characteristic Area Under Curve* (AUC) score. The Receiver Operating Characteristic (ROC) curve is a plot comparing true positive rate against the false positive rate, from which we derive a single statistic: AUC, the integrated area of the ROC curve. In the comparison of hidden layer representation, Raghu et al. invoke *Singular Vector Canonical Correlation Analysis* (9) (CCA). The goal of this work is to measure the performance difference between the reports in the Raghu paper and our own implementations of their work. Therefore, we have generated our own results and compare them qualitatively to their results.

7 Results

We evaluate the performance of the state-of-the-art models on the Retina task, comparing the AUC performance of both transfer and random initialization in table 1. The evolution of the AUC performance for all runs can be seen in figures 1 and 2. We also calculate the CCA performance for for the models, comparing the representations in the final hidden layer of models trained from random initialization and from transfer initialization, using the similarity of two random init models as a baseline. This comparison can be see in figures 3 and 4.

Model	**Random Init AUC**	**Transfer AUC**
Resnet-50 (theirs)	96.4% $\pm$ 0.05	96.7% $\pm$ 0.04
Resnet-50 (ours)	81.25 % $\pm$ 0.02	86.89% $\pm$ 0.03
InceptionV3 (theirs)	96.6% $\pm$ 0.13	96.7% $\pm$ 0.05
InceptionV3 (ours)	86.57 % $\pm$ 0.03	93.1 % $\pm$ 0.007

Table 1: Performance of models on Retina Data. Note the large deviation, both in AUC and in the different impact of transfer initialization between our ResNet models and theirs. This is accounted for by the difference dataset scale described above. While our results seem to reinforce the 'Very Small Data Regime' result as mentioned above, it contradicts the results of the extended results given in their paper's appendix: the improvement seen in models with transfer over random initialization is not anticipated by their work.

7.1 Replicability Analysis

Our results do not align easily with those of Raghu et al. We forward a several possibilities that could account for this, including differences in dataset size, differences in data quality (both of images and labels), and transfer initialization realization.

In spite of the difference in performance, we do no feel that our results are sufficient to contradict any of their results.

[1]model metrics available here:https://pytorch.org/docs/stable/torchvision/models.html#classification

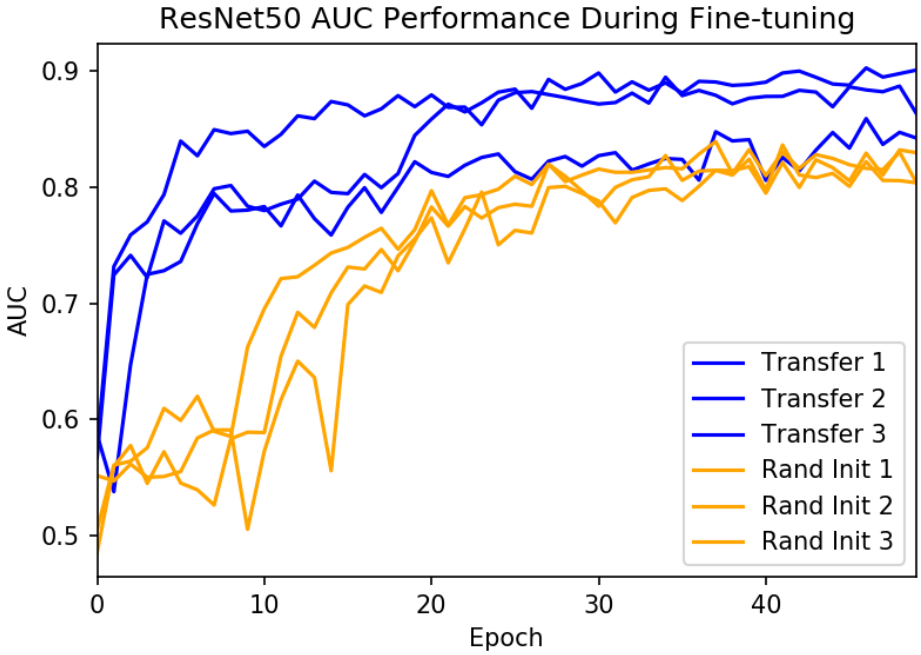

Figure 1: The epoch evolution of each of the ResNet models' AUC for the validation dataset. Notice that the transfer-init models improve faster than the random-init models, however the overall performance difference is not as extreme after all models converge.

8 Conclusion

We compared performance of the ResNet models with different initialization methods. We found differences in performance which emphasise the usefulness of transfer initialization outside of the presence of massive in-task-domain data sets. We were unable to substantiate the claim that the difference in distribute from natural images to medical images marginalizes the usefulness of transfer learning for medical imaging tasks.

8.1 Future Work

As mentioned in the Replicability Analysis section, transfer initialization realization quality plays a role of unknown magnitude for transfer learning tasks. Neither this work nor theirs considered the impact of transfer weight initialization, and it may be the case that some pretrained models are less suited for transfer than others. Addressing this will require a more rigorous layer-wise correlation analysis to find when models stop developing generalizing features and begin building task-focused features.

The authors also believe that more work on the mean-var transfer initialization from the raghu paper will be informative insofar as explaining which aspects of transfer initializations result in higher-performance on the transfer task.

References

[1] M. Raghu, C. Zhang, J. M. Kleinberg, S. Bengio, Transfusion: Understanding transfer learning with applications to medical imaging, CoRR abs/1902.07208. `arXiv:1902.07208`.
URL `http://arxiv.org/abs/1902.07208`

[2] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, A. Y. Ng, Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison (2019). `arXiv:1901.07031`.

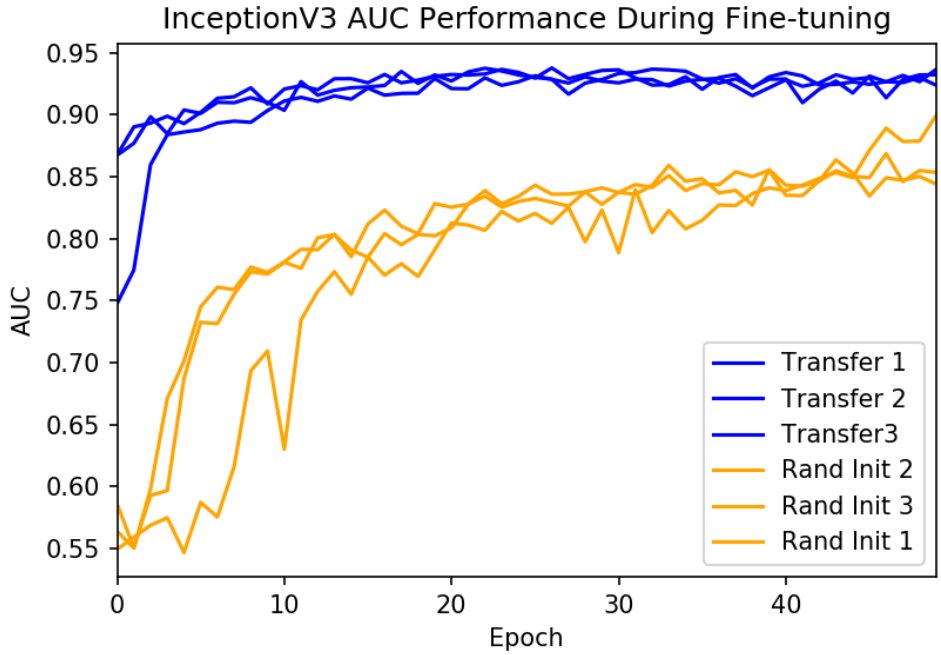

Figure 2: The epoch evolution of each of the Inception models' AUC for the validation dataset. Unlike the ResNet models, while we see again that the transfer-init models converge faster, this time the random-init models do not achieve the same degree of performance.

[3] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. C. Nelson, J. L. Mega, D. R. Webster, Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs, JAMA 316 (22) (2016) 2402–2410. arXiv:https://jamanetwork.com/journals/jama/articlepdf/2588763/joi160132.pdf, doi:10.1001/jama.2016.17216.
URL https://doi.org/10.1001/jama.2016.17216

[4] F. Olshausen, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (1996) 607–609. arXiv:https://www.nature.com/articles/381607a0#citeashttps://www.nature.com/articles/381607a0#citeas, doi:10.1038/381607a0.

[5] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CoRR abs/1512.03385. arXiv:1512.03385.
URL http://arxiv.org/abs/1512.03385

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, CoRR abs/1512.00567. arXiv:1512.00567.
URL http://arxiv.org/abs/1512.00567

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252. doi:10.1007/s11263-015-0816-y.

[8] Diabetic retinopathy detection kaggle competition.
URL https://www.kaggle.com/c/diabetic-retinopathy-detection/data

[9] M. Raghu, J. Gilmer, J. Yosinski, J. Sohl-Dickstein, Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability (2017). arXiv:1706.05806.

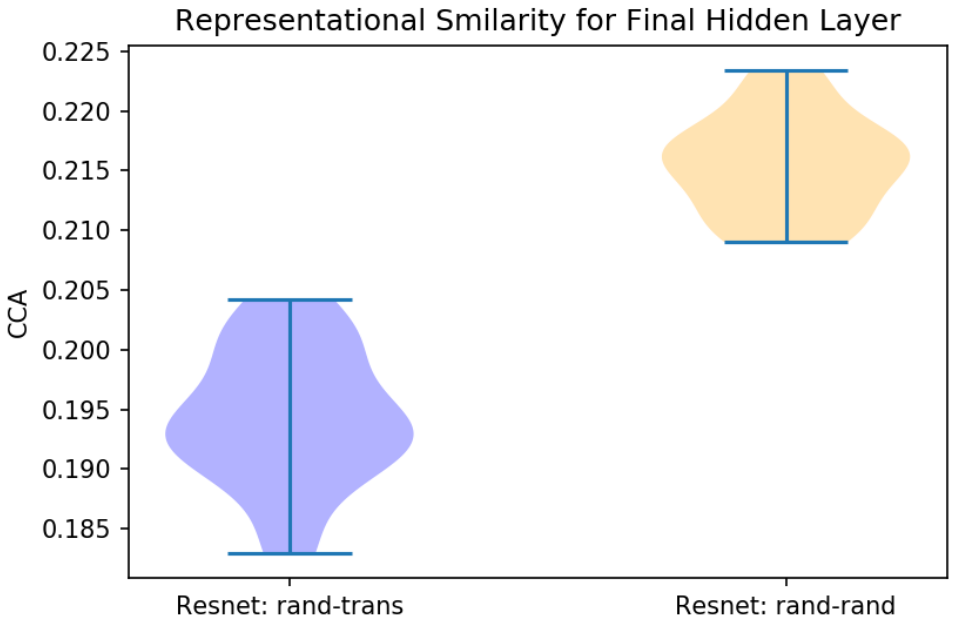

Figure 3: This comparison shows the CCA similarity of the activation vector of the final layer of a transfer-init model and a random-init model, with a baseline of two random-init models' similarity. This is meant to show that the transfered model is less similar to a model with random initialization than two randomly initialized models are similar to one another, and so justify the claim that the transfered model learns new representations from the randomly initialized model.

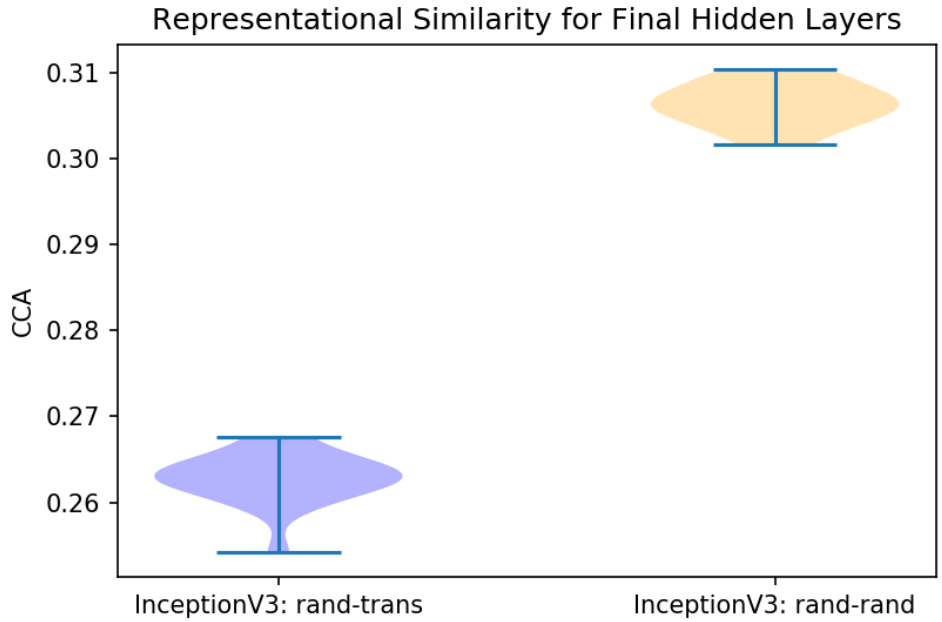

Figure 4: Unlike the CCA comparison for ResNet, the Inception models display a greater gap in similarity, more akin to the figures in their paper. This may be confounded by the performance gap noted in table 1.