Representational Similarity Analysis vs Performance

Florian Bemmerl [1] **Aleksandr Chebykin** [1] **Yang Chen** [1] **Janek Willeke** [1]

Abstract

Representational Similarity Analysis (RSA) is a method from neuroscience used to compare representations of different models. While it has been applied to neural networks before, there exists no comprehensive study on how representation as measured by RSA relates to performance. We relate model accuracy to RSA on FashionMNIST and ImageNet datasets.

1. Introduction

Representational Similarity Analysis as introduced by (Kriegeskorte, 2008) is a two-step procedure that allows comparing representational spaces. For a set of inputs, e.g. images, RSA first computes how dissimilar inputs are within a model's representational space. Then, to compare models, RSA compares these representational dissimilarities.

RSA, therefore, is an indirect way of comparing model representations, which can be used when directly comparing representations is not possible. In neuroscience, RSA is used to compare representational spaces of models of the brain to the representational spaces of the brain itself.

For neural networks of different architectures representational space can be very different. Thus, RSA can provide means to compare these spaces. The literature on neural networks and RSA has so far focused on comparing CNNs to the (human) brain, see e.g. (Kietzmann et al., 2017). However, there exist a few papers that apply RSA outside of a neuroscience context, compare (Dwivedi & Roig, 2019) or (Nayebi & Ganguli, 2017). Yet none of them focus on how representation as measured by RSA and model performance as measured by accuracy relate. Intuitively there should be a strong relation, as a model's prediction is dependent on it's internal representation. We perform RSA on models of varying performance levels trained on both FashionMNIST and ImageNet.

Our code is available at `https://github.com/AwesomeLemon/NI-Project-RSA`.

[1]Faculty IV, Technical University of Berlin, Berlin, Germany.

NI Project, Summer Term 2019, Berlin, Germany, 2019.

2. Representational Similarity Analysis

2.1. Methods

To perform RSA on artificial neural networks we set up various models using the PyTorch framework (Paszke et al., 2017) and compute the model activations on different layers for multiple inputs. Also we relate models to each other by computing representational similarities between them, using the following methods.

Input RDM For each image in a fixed set of inputs we compute the model activations of one layer. For each pair of activations we calculate the (Pearson) correlation. Arranging the values associated with the two inputs yields a matrix symmetric about a diagonal of zeros. As in (Kriegeskorte, 2008) we take $1 - corr$ for every activations pair to derive a Representational Dissimilarity Matrix (RDM) on the inputs. Input RDMs can be used to compare representations of different models, as seen in Fig 1 on a small representative sample of the FashionMNIST dataset, that is used throughout this section. Taking the activations of the penultimate layer for the four shown samples, the Input RDM of the high performing model shows clear dissimilarity regarding the first item to the others.

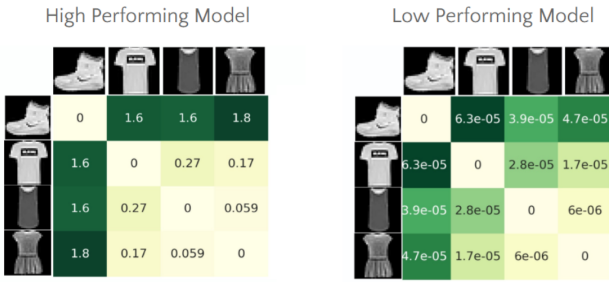


Figure 1. Input RDMs of two FashionMNIST models with different performances on same four samples

Model RDM For each model pair we compute the dissimilarity between their Input RDMs using (Spearman) correlation or euclidean distance. Model RDM is the matrix of pairwise dissimilarities of Input RDMs, as shown in Fig 2. There, four Input RDMs of networks with different performances were calculated on the same FashionMNIST dataset and pairwise correlations calculated using rank correlation.

The model accuracies range from low to high from left to right. Intuitively, the representations of low performing models are more dissimilar from the representations of high performing ones, in comparison to representations of the models from the same performance range.

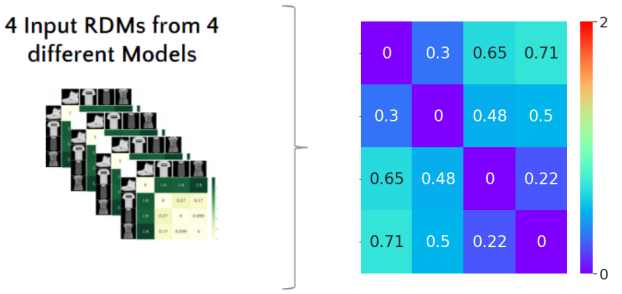


Figure 2. Model RDM of four FashionMNIST models (on same inputs) using rank correlation distance

Confusion Matrix For a classification task, a confusion matrix is a specific table representation that allows visualisation of the performance of an algorithm. Entries in the matrix represent the instances of a predicted class for an actual class, therefore showing if a system is confusing two classes (i.e. mislabelling them).

A confusion matrix is the most common (yet of course superficial) tool to gain insights into the inner workings of neural networks. Therefore we will compare it to RSA, when appropriate. In section 4 we consider the extended top-5 confusion matrix as well. It extends the class confusion up to the 5 most predicted classes, as in the commonly used top-5 model error for classification.

Model Architectures We trained 562 models on FashionMNIST. They fall into two categories: MLPs and CNNs. Within every category, the depth and order of layers is the same. However, the size of each layer is varied. The maximum number of hidden layer nodes is 45 for MLPs. For CNNs, there is a maximum of 24 filters used in the convolutional layer, with a maximum of 192 units after the first fully connected layer. No two models share the same architecture.

Table 1. Sequence of layers of MLP and CNN models

Layer	MLP	CNN
1	FC[1], Sigmoid	Convolutional, Max-pool, ReLU, Batch-Norm, Dropout
2	FC, Sigmoid	FC, ReLU, Batch-Norm, Dropout
3	FC, Sigmoid	FC, ReLU, Batch-Norm, Dropout
4	FC, Softmax	FC, Softmax

MLPs range in performance from 19.20 % to 81.69 % accuracy, CNNs from 79.35 % to 89.19 % accuracy, with 90 % of CNNs having accuracies greater than 84.88 %.

[1]Fully Connected

2.2. Distance function for Model RDM

In Fig 2 we used rank correlation for constructing the Model RDM, but it's actually not the preferred metric on the second level if model accuracy is important to the analysis. Taking the two models and their Input RDMs from Fig 1, one sees how colour intensities follow the same pattern, which means that Input RDMs correlate highly, although the magnitudes of values are vastly different. Between these two we get a rank correlation distance of 0.1 out of 2.0 maximum and a euclidean distance of 2.9 out of a 4.9 maximum. This example shows the difference of the chosen dissimilarity measure between Input RDMs.

While rank correlation works for the neuroscience means of (Kriegeskorte, 2008), it ignores magnitudes. But low-magnitude distances between representations mean that a model is susceptible to noise, which cuts down a classifier's performance. As euclidean distance accounts for magnitude, it is a more suitable measure for constructing Model RDMs of neural networks.

As seen in Fig 3 the difference between using rank correlation (3a) and euclidean distance (3b) gets more striking in case of analysing Model RDMs for a multitude of models. When comparing Model RDMs of all 562 models on FashionMNIST, euclidean distance lets us differentiate between low- and high-performing models (accuracy is sorted from top to bottom). While with rank correlation high- and low performing models are not clearly distinguishable, they are clearly dissimilar in euclidean space. Low-performing models may be both very similar or dissimilar but with increasing performance model similarity increases visibly in Fig 3b. High-performing models are very similar, especially CNNs among themselves.

Using Manhattan Distance as the distance measure would be suitable as well, since it also accounts for magnitudes and yields the same result as euclidean distance.

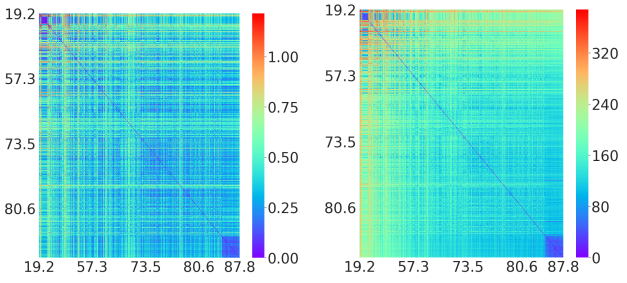


Figure 3. **Model RDM** with a) **Rank correlation** and b) **Euclidean distance** of 562 models sorted by accuracy (increasing)

2.3. RSA of Earlier Layers

In our other experiments, we perform representational similarity analysis on the penultimate layers of various artificial neural networks. However, we also want to investigate the effects of RSA for layer-pairs at different depths. Thus, we have also calculated the model representational dissimilarity matrices for three earlier hidden layers for all MLPs and CNNs that we have built and trained. To reveal the representational dissimilarity changes over layers, we just column-wise average the corresponding three Model RDMs for each layer.

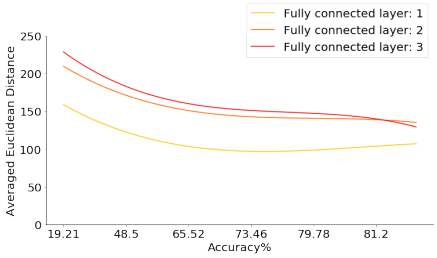


Figure 4. Representational dissimilarity over earlier layers

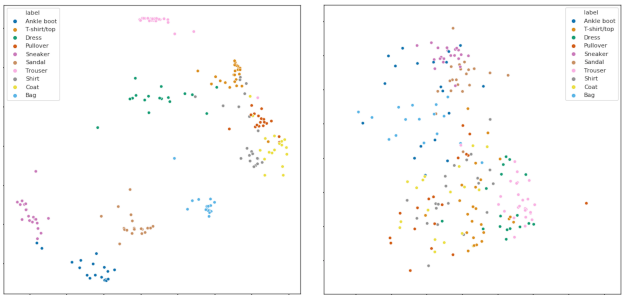
After smoothing, as depicted in Fig 4, the hue of the curves indicates the depth of the corresponding layer. Moreover, each value in accuracy-axis represents one neural network. Hence, given a model with this accuracy, the average dissimilarity of a model with all the other models increases as we go deeper into the network. Moreover, in any hidden layer (for instance, in the first fully connected layer) the Input RDMs of low-performing models are very dissimilar with those from the high-performing models.

Intuitively, for any pair of neural networks, the Input RDMs of the third hidden layers should be more dissimilar than those of the first hidden layers (since models should learn similar basic features in earlier layers, and specialise in later layers). However, there exist some exceptions in Fig 4, where the accuracy is beyond 82%. The models in this range are overall CNNs. One possible interpretation could be that CNNs capture different features respectively at first, and they converge finally to a similar data representation in the end: from diverging to converging. Moreover, another interpretation could be that the observed difference is not significant: maybe for the second and third to last layers of CNN, the representations of data are just very similar because the difference lies in the convolutional layer.

3. Structures in Representational Space

On the Model RDM, we apply the multidimensional scaling (MDS) embedding method to generate 2D-visualisation plots, which allow us to observe whether RSA could differentiate between models regarding their performances.

MDS projects pairwise distances (dissimilarities) of a set of points onto a lower-dimensional space, aiming to preserve the between-object distances as well as possible.

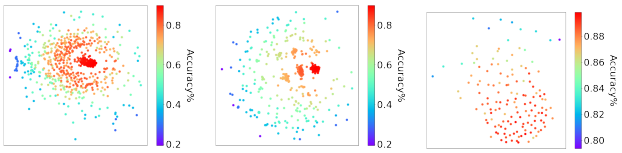


(a) (b)

Figure 5. **MDS embedding** of penultimate layer activations on a mini-batch from a **(a) high-performing** model and a **(b) low-performing** model

In a model with higher accuracy, samples from the same class should be represented similarly, while the distances between dissimilar classes should be large. That is, the embeddings of a high-performing model should be more structural, so that it could perform classification effectively, as depicted in Fig 5a. In contrast, samples are mixed up in a low-performing model (Fig 5b).

One of our goals is to reveal how models with different performance levels relate to one other. Thus, similar to visualising how a model represents its input in embedding space by considering pairwise distances between samples, we can interpret Model RDM as containing pairwise distances between the models, and visualise them in the same way. The embedding result is shown in Fig 6a.



(a) (b) (c)

Figure 6. **(a) MDS embedding** of **(a) Input RDMs** and **(b) confusion matrices** of all trained models; **(c) MDS embedding** of all CNNs' Input RDMs

With MDS embedding representational differences between high and low performing models become apparent. All CNNs cluster densely in the centre of Fig 6a, which implies that they encode the input similarly. Moreover, there exists a gap between this central cluster and its surrounding points(MLPs). One possible reason is that almost 90% of CNNs are no less than 3 percentage points better than even the best MLP. Another explanation could be that CNNs and

MLPs arrive at different representation classes due to their structural differences.

Instead of only performing MDS on Input RDMs, we calculate the corresponding confusion matrices for each model as well. Again, we perform MDS on all confusion matrices as Fig 6b, which gives a similar result as Fig.6a. That is, all CNN models fall into the centre of this embedding space surrounded by MLPs. Thus, we would stress that Input RDM is similar to be a higher resolution confusion matrix. More specifically, Input RDM shows sample-wise dissimilarities, and the confusion matrix is an interclass comparison.

If we zoom into the centre of Fig 6a and observe only CNN models as depicted in Fig 6c, they appear to form a gradient of performances, from the upper left corner to the lower right part. It would allow us to differentiate even high-performing models using MDS.

4. RSA on ImageNet

Having performed variable experiments with the models trained on FashionMNIST, we transfer gained insights to analysis of models trained on ImageNet LSVRC 2012 (Deng et al., 2009). Imagenet is a bigger and more challenging dataset used for benchmarking research progress in image classification.

4.1. Technical information

Due to the size of ImageNet (more than 14 million images), we could not reasonably train the same number of models as on FashionMNIST, since we lacked necessary computing power in the form of GPUs. Instead we had to rely on pre-trained models, which all perform well ($> 55\%$ top-1 accuracy; see Table 2 for details of used models and their performances). This restriction makes analysis harder, as there are no more stark contrasts arising due to extensive performance differences. But on the positive side, this situation is better suited for testing usefulness of RSA, because theoretically RSA should be utilized exactly in the conditions of a complicated dataset and availability of well-performing models only, which should be meaningfully analysed and compared using the technique.

We used PyTorch and pretrained models shipped with it.

4.2. Input RDMs and confusion matrices

In section 3 we show that on the FashionMNIST-trained models confusion matrices produce almost the same MDS embedding as Input RDMs (of the activations of the penultimate layer). It seems that both Input RDMs and confusion matrices on FashionMNIST contain the same information.

[3] https://pytorch.org/docs/stable/torchvision/models.html

Table 2. Pretrained models used. Every architecture type is referenced only once. Accuracy as reported in pytorch documentation[3].

Model name	Top-1 accuracy, %
AlexNet (Krizhevsky et al., 2012)	56.55
SqueezeNet 1.0 (Iandola et al., 2016)	58.10
SqueezeNet 1.1	58.19
ResNet-18 (He et al., 2016)	69.76
GoogleNet (Szegedy et al., 2015)	69.78
VGG-13 (Simonyan & Zisserman, 2014)	69.93
VGG-11 (batch-normalized)	70.38
VGG-13 (batch-normalized)	71.55
VGG-16	71.59
MobileNet V2 (Sandler et al., 2018)	71.88
ResNet-34	73.30
VGG-16 (batch-normalized)	73.37
VGG-19 (batch-normalized)	74.24
Densenet-121 (Huang et al., 2017)	74.65
ResNet-50	76.15
ResNet-101	77.37
ResNeXt-50-32x4d (Xie et al., 2017)	77.62
ResNeXt-101-32x8d	79.31

Confusion matrices, however, are much less computationally intensive to calculate and therefore preferable if they indeed contain the same information as Input RDMs. But it might be that the additional information in Input RDMs is not apparent in a simple dataset like FashionMNIST.

In Fig 7 we compare an Input RDM, top-1 and top-5 confusion matrices of the activations of the penultimate layer of DenseNet-121.

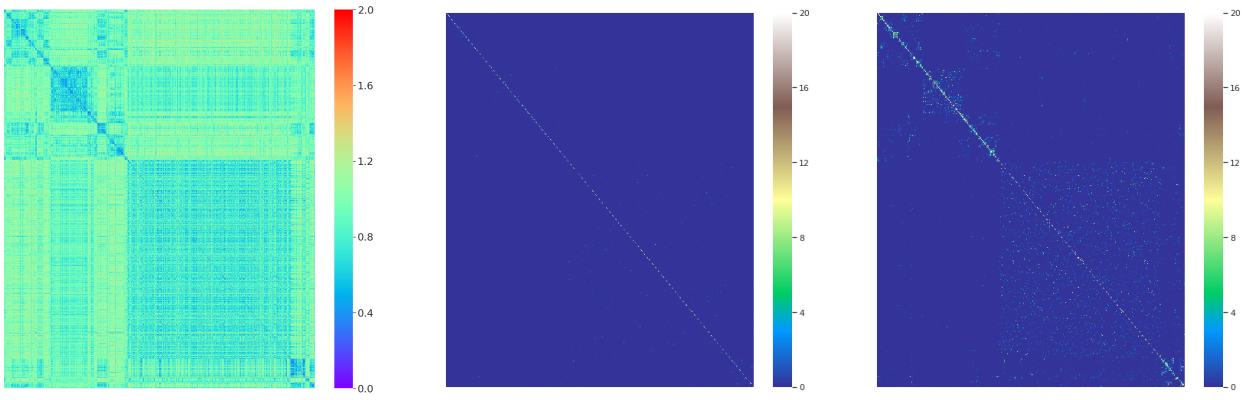
While in Input RDM sizeable blocks of samples similar between themselves are clearly visible (hinting at similar representational structure), the top-1 confusion matrix shows little besides a well-pronounced diagonal, which is to be expected from a well-performing model (because it means that the majority of samples is classified correctly).

Top-5 confusion matrix, on the other hand, seems to hint at the same structure of blocks, which are however far less-pronounced. To find out if there is a more quantifiable difference between Input RDMs and top-5 confusion matrices, we investigate MDS embeddings of both.

4.3. Embeddings and different representation types

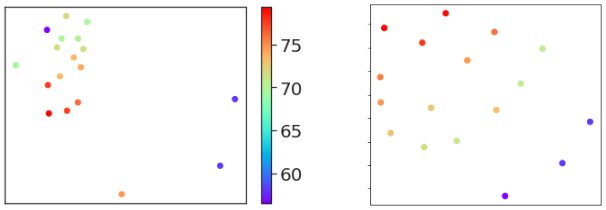
Resulting embeddings are presented in Fig 8. In both images we see a colour gradient: models are arranged from low-performing ones over middle-performing to high-performing, with few outliers. But in the embedding produced by confusion matrices models are almost equidistant and seem to be simply sorted by performance.

In the embeddings of Input RDMs, to the contrary, models are not evenly spread-out. Instead we see a cluster, sorted more or less by performance, and several outliers.



(a) (b) (c)

Figure 7. **(a) Input RDM** of DenseNet-121 on 10,000 samples (10 per class) of ImageNet Validation Set; **Confusion matrices** of **(b) top-1** and **(c) top-5** error of the 1,000 classes. Samples used for calculating Input RDM are sorted by class. The order of classes is the same as in confusion matrices.



(a) (b)

Figure 8. MDS embedding of **(a) Input RDMs** and **(b) confusion matrices** of the ImageNet-trained models

When we examine Input RDMs by hand, an interesting property starts to emerge. Input RDMs of models in the cluster have close-to-diagonal structure (see AlexNet and ResNeXt-101-32x8d in Fig 9). This means that only representations of objects from the same class are similar, and they are mildly dissimilar to representations of objects belonging to different classes. The best-performing model, ResNeXt-101-32x8d, visually has the most pronounced diagonal structure. This behaviour is expected: well-performing models learn to differentiate every class from every other class.

But examining 3 outlying Input RDMs leads to another picture. Sizeable block structures become visible in all the outliers, DenseNet (Fig 7a) and SqueezeNets 1.0 and 1.1(Fig 9c). Bearing in mind that the depicted heatmap is similarities of 10,000 times 10,000 samples, we see that representations of objects in many classes are similar among themselves and strongly dissimilar to representations of objects from some other large group of classes. It is counter-intuitive that such a structure of representation similarities should lead to good performance, and yet it does. DenseNet-121 (Fig 7a) is the fifth-best-performing model used by us, which means that performance-wise it leaves behind many models with diagonal Input RDMs that find themselves in the tail end of the cluster.

The other 2 outliers, SqueezeNets, are also an interesting case. They display even more pronounced blocks of similarity and dissimilarity (SqueezeNet 1.1 is depicted in Fig. 9c, yet they perform worse than DenseNet-121.

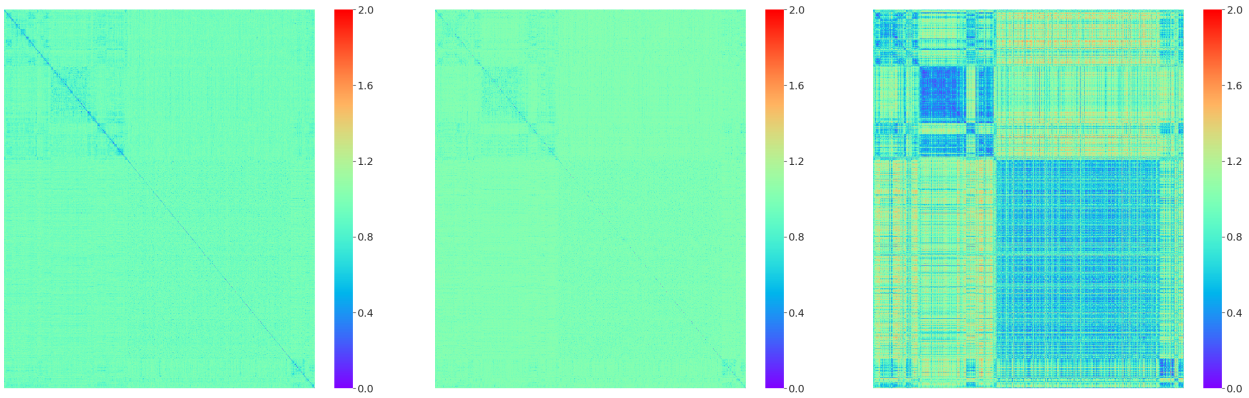
Evidence in the form of MDS embeddings seems to point that an Input RDM contains more information than even the top-5 confusion matrix: while embedding of confusion matrices seems to just sort models by accuracy, putting AlexNet and SqueezeNets nearby, in embedding of Input RDMs a distinction between different representation types seem to be visible, with AlexNet (Input RDM of which has diagonal structure) placed far apart from SqueezeNets (Input RDMs of which have the structure of blocks), while SqueezeNets are close to each other.

Possible implications of these representational differences are discussed in Section 5.

5. Conclusion and future work

Using euclidean distance as a measure of dissimilarity, we can show that models of the same performance level have similar representations. Rank-correlation as a dissimilarity metric obfuscates this relation, as it does not account for magnitude of representational dissimilarity.

Our findings suggest that computing the Input RDM of the penultimate layer of a neural network yields similar information about a models representation as a confusion matrix. Whereas confusion matrices measure model confusion on a per-class basis, Input RDMs show dissimilarity on a per-sample basis. Thus the information provided by



(a) ResNeXt-101-32x8d, 79.31% (b) AlexNet, 56.55% (c) SqueezeNet 1.1, 58.19%

Figure 9. Input RDMs and top-1 accuracies of selected models

Input RDMs is more granular. In FashionMNIST, this granularity does not give rise to new insights. On ImageNet, however, it appears that with RSA we gain deeper insights into representational structure than with a confusion matrix.

One application of our findings lies in transfer learning. RSA on ImageNet models suggests that there are models of similar performance level, yet different structures of representational dissimilarity. When choosing a model pretrained on ImageNet for a new task, a certain structure of representational dissimilarity might be preferable. Consider e.g. a new task of predicting whether an input is a fish or a bird. Some of the ImageNet classes are fish, some of them bird. When picking a model pre-trained on ImageNet, we want a model that has learned that e.g. a goldfish is more similar to a sturgeon than it is to a hen. A model pre-trained on ImageNet with a blocky structure might be able to pick up on shared similarites within classes of fish and classes of bird. It might be better suited for our specific task of differentiating between fish and birds, as its representations of them are more refined in comparison to a model with diagonal Input RDM. RSA could thus offer guidance in choosing a pre-trained model.

Moreover, RSA might prove useful as a performance metric in unsupervised learning, where there exists no general performance metric. If RSA performed on unsupervised models gives rise to something that similar to a confusion matrix, it could be used to compare performance of unsupervised models of arbitrarily dissimilar architectures.

References

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Dwivedi, K. and Roig, G. Representation similarity analysis for efficient task taxonomy transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

Kietzmann, T. C., McClure, P., and Kriegeskorte, N. Deep neural networks in computational neuroscience. May 2017. doi: 10.1101/133504. URL https://doi.org/10.1101/133504.

Kriegeskorte, N. Representational similarity analysis – connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2008. doi: 10.3389/neuro.06.004.2008. URL https://doi.org/10.3389/neuro.06.004.2008.

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

Nayebi, A. and Ganguli, S. Biologically inspired protection of deep networks from adversarial attacks. *arXiv preprint arXiv:1703.09202*, 2017.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.