Refining Deep Neural Networks via Interpretability Using Transfer Learning

Xiaoguang Wang
Dimension Institute
Hamilton, Ontario, Canada
xwang@dimensioninstitute.org

Xuan Liu
The Hospital for Sick Children (SickKids)
Toronto, Ontario, Canada
xuanv.liu@utoronto.ca

ABSTRACT

Current research on interpretability tend to focus on building interpretable models for highly non-interpretable neural nets. Little work has been done on employing interpretability for refining models. We propose to leverage transfer learning to refine deep neural nets. Combined with a contemporary data visualization technique for interpretation, we are able to show empirically why VGG19 has better classification accuracy than Alexnet on the CIFAR-10 dataset through quantitative and qualitative visualizations on each of the hidden layers. This approach could be applied to refine neural nets when altering the parameters of the hidden layers for deep neural nets. Compared with a previous approach which just apply neural feature visualization, we are able to show not only qualitatively but also quantitatively why one model has higher accuracy than another one. Compared with Knowledge Distillation, we directly interpret the complex model using transfer learning via a simpler dataset, without distilling the complex model into a shallow one.

ACM Reference Format:
Xiaoguang Wang and Xuan Liu. 2024. Refining Deep Neural Networks via Interpretability Using Transfer Learning. In *Proceedings of 33rd ACM International Conference on Information and Knowledge Management (CIKM '24).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

With the fast development of sophisticated machine learning algorithms, artificial intelligence has been gradually penetrating a number of brand new fields with unprecedented speed. One of the outstanding problems hampering its further progress is the interpretability challenge. This challenge arises when the models built by the machine learning algorithms are to be used by humans in their decision making, particularly when such decisions are subject to legal consequences and/or administrative audits. For human decision makers operating in those circumstances, to accept the professional and legal responsibility ensuing from decisions assisted by machine learning, it is critical to comprehend the models. This is generally true for areas such as criminal justice, health care, terrorism detection, education systems and financial markets.

Deep Neural Networks (DNNs) have revolutionized the field of artificial intelligence with their unparalleled ability to perform complex tasks across various domains, including image recognition, natural language processing, and medical diagnosis. Despite their success, DNNs are often criticized for their lack of interpretability, which poses significant challenges in critical applications where understanding the decision-making process is essential.

Interpretability in deep learning refers to the ability to explain or to present in understandable terms to a human how a model makes its decisions [15]. The demand for interpretability arises from the need for transparency, accountability, and trust in AI systems, especially in high-stakes domains such as healthcare and criminal justice. Interpretability is not only crucial for validating the model's reasoning but also for identifying biases, ensuring fairness, and facilitating the improvement of the model by understanding its shortcomings.

Interpretability can serve as a powerful tool for refining DNNs. By understanding the model's decision-making process, researchers and practitioners can identify and correct errors, remove biases, and improve the model's generalization capabilities. For instance, biology-inspired deep learning models, which incorporate biological knowledge into their architecture, have shown that interpretations can vary and be influenced by biases in the underlying knowledge [22]. Addressing these biases can lead to more robust and reliable interpretations, ultimately improving the model's performance.

Moreover, interpretability can enhance the adversarial robustness of DNNs. Regularizing input gradients to improve interpretability has been shown to increase the model's resistance to adversarial attacks [28]. This suggests that interpretability and robustness may be interconnected, and focusing on one can benefit the other.

Speaking of interpretability, we should be cautious that the meaning is two-fold: one from the perspective of the end users and the other from the perspective of the model designers, which demand different explanations and measures of efficacy. For end users, it is mainly employed to illustrate predictions in unforeseen circumstances and build a sense of trust. For model designers, it is useful to diagnose and refine the models. Current research [14, 17, 21, 27, 33] tend to focus on learning interpretable models, but these models are seldom leveraged to help diagnose [40] and refine the non-interpretable complex models.

Recent research has made significant strides in developing methods to interpret and explain the decisions of DNNs. Interpretation

CIKM '24, October 20–25, 2024, Boise, Idaho, USA

ACM ISBN 978-1-4503-XXXX-X/24/10...$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

tools aim to shed light on the black-box nature of these models, providing insights into their decision-making processes [15]. A comprehensive survey by Li et al. (2022) [15] introduces two fundamental concepts: interpretations and interpretability, which are often conflated. Interpretations are the explanations provided for individual decisions, while interpretability is the overall characteristic of a model that describes how easy it is to understand its workings.

A taxonomy of interpretation algorithms has been proposed, categorizing them based on different perspectives, such as the type of explanation they provide (e.g., local vs. global) and the techniques they employ (e.g., perturbation-based, decomposition-based). Performance metrics for evaluating these algorithms have also been surveyed, which is crucial for assessing the effectiveness of interpretation methods.

In this paper, we demonstrate from the angle of the model designers on how interpretability could help improve a model's accuracy. The research that most close to ours is a feature visualization approach proposed in [38] that employs visual interpretation to diagnose the problems of a already existing deep learning model: Alexnet [13] to refine it. They utilizes a multi-layered Deconvolutional Network (Deconvnet) [39] (initially designed for unsupervised learning), which maps the feature activities back to the input pixel space and could find the optimal stimulus at any hidden layers in the model. After visualizing the first and second hidden layers of the Alexnet, they reduced the filter size of the first hidden layer from 11 × 11 to 7 × 7 and the stride of convolution from 4 to 2. The resulting model outperforms the architecture of Alexnet for their single models by 1.7% (test top-5).

However, the justification / intuition for the choice of smaller filters wasn't convincing enough. Their conclusion relies on the vague differences of visualizations. They changed the parameters of the first hidden layer, but when we examine the corresponding visualizations in the first hidden layer we don't see much differences visually. In this paper, we propose a method to quantitatively measure the visualizations on each hidden layer.

Our research focus on developing methods that enhance interpretability without compromising the model's accuracy. One related promising approach is the use of symbolic expressions to add interpretability to already-trained models [18]. This method fits symbolic expressions to the functions within the model, potentially offering a way to maintain high accuracy while providing interpretable predictions.

Another related research to ours is Knowledge Distillation [7] [5] [33], which refers to the process of distilling the dark knowledge learned by a teacher model (usually sophisticated and large) to a student model (usually shallow and small). Different from the Knowledge Distillation technique, our approach interprets the deep learning model directly through transfer learning without distilling it into a shallow neural network.

The contributions of this paper are several fold: first, feature visualization and data visualization are two separate visualization techniques, we are the first to combine them together to refine neural network models more convincingly; second, we are also the first to use visualization techniques for interpreting the transfer learning results on CIFAR-10 dataset; third, we propose a way to quantity interpretation results via transfer learning; fourth, unlike knowledge distillation, we interpret deep learning models directly without distillation; finally, we are also the first to compare the results of two different visualization methods (feature visualization and data visualization) on the same dataset.

2 METHOD

There are a number of interpretation methods proposed for neural networks, which could be roughly categorized as post-hoc interpretations, inherently interpretable models and other models such as Influence Functions (IF) [12], SHapley Additive exPlanations (SHAP) [19], Information Bottleneck (IB) [31], etc. Post-hoc Interpretations [21] [6] interprets black box models after they are trained, hence, the name "Post-hoc". Usually, this requires the building of a separate interpretation model or technique to explain the predicted decisions or the model itself. The majority of the interpretation methods belong to this class and here we roughly divide them into four subcategories: Interpretation by Perturbation [30][21], Local Interpretations [27], Global Interpretations [14] [2] [33] and Visualization. Among visualization techniques, there are neural feature visualization, attribution and data visualization [25]. In this paper, we compare the results of feature visualization with data visualization for refining deep neural nets.

2.1 Neural feature visualization

With the thriving progress made in the past few years, feature visualization has established itself as the most promising research direction for neural network interpretations.

Usually, the most commonly applied technique is Activation Maximization (AM) [4]. This method enables the interpretation of arbitrary layers of a neural network, not just the first layer representation (linear weights in the input-to-first layer weight matrix) that could be easily interpreted by the learned filters . It also assumes that the input data are meaningful and displayable for humans, e.g. image data.

The idea of Activation Maximization is remarkably simple, but could generate high-quality visualizations. It essentially searches for input patterns which maximize the activation of a given hidden unit. This works because the patterns which fire the maximum activation could be a good first-order representation of what a unit is doing. This idea could be formulated as an optimization problem:

$$x^* = \arg\max_x h_{ij}(\theta, x) \tag{1}$$

Here x^* is the optimal input pattern that the method tries to find and h_{ij} stands for the activation at unit i from a given layer j of the previous layers and θ represents the parameters of model. For an already trained neural network, these parameters are known. The maximum of h_{ij} is found by calculating the gradient of $h_{ij}(\theta, x)$ and moving x in the direction of this gradient. This step is called gradient ascent.

However, there are two shortcomings to this approach. First, it is hard to do initialization. It was mentioned that different random initializations sometimes generate the same optimal stimulus [4]. Second, the information about the invariance (the range of inputs that the unit is invariant to) is not available from the optimal stimuli which is just a single image. To address the second disadvantage of AM, the creators of Tiled convolutional neural networks (TCNN) [23] applied the method in [1] and extend it to arbitrary networks in order to visualize the invariant directions of a hidden unit [1]. However, the output of hidden neurons for TCNN are non-quadratic functions of inputs while [1] studies quadratic functions. This makes the extremely complex invariance of TCNN hard to be precisely captured.

Hence, another visualization approach [38] which utilizes a multi-layered Deconvolutional Network (Deconvnet) [39] (initially designed for unsupervised learning) is proposed to find non-parametric views of invariances. This approach maps the feature activities back to the input pixel space and could find the optimal stimulus at any hidden layers in the model. This method is also a successful case that employs visual interpretation to diagnose the problems of a already existing model [13] to improve the results. Therefore, in our paper we adopt Deconvnet as our neural feature visualization method.

2.2 Data visualization

Understanding data by visualizing them is an intuitive and important approach. Plotting two or three dimensional data is an easy task for most graphing tools. But for data that has more than three dimensions, special techniques are needed to transform them into a more visually understandable two-dimensional space. These techniques are called Dimension Reduction [36]. They could also be potentially helpful for assisting the interpretation of black box models.

Some of the popular dimension reduction techniques are Principal Component Analysis (PCA) [10], Multidimensional Scaling (MDS) [35] [3], t-distributed Stochastic Neighbor Embedding (t-SNE) [20] and Autoencoder networks [9]. Among these, t-SNE has become the de facto standard for a variety of applications. t-SNE mitigates the two problems that SNE [8] has: the optimization problem and the 'crowding problem'. It is able to reveal the local structure of the data as well as the global structure (such as clusters at multiple scales). And it also generates significantly better visualizations which was demonstrated in experiments [20] by comparison with many other non-parametric visualization techniques such as Sammon mapping, Isomap, and Locally Linear Embedding.

Recently, there are some applications of applying dimension reduction for interpretability. Two dimensional embedding is generated in [11] applying t-SNE on the hidden layers of Alexnet. t-SNE is also applied for reinforcement learning in [37]. In [24], PCA and t-SNE are employed combined with the k-means algorithm for the purpose of different facet visualizations. Similarly, a dimension reduction technique (not specified in the paper) is applied in [34] to present the final visualization of the treeview method they proposed for peeking into the classification process of a multi-layer perceptron. In this paper, we plan to quantify interpretation results utilizing t-SNE.

2.3 Compare Deconvnet and t-SNE to refine deep convnets

Neural feature visualization and data visualization are two separate interpretation strategies. In the past few years, they develop along each of their own paths, but never were combined for interpretation. Neural feature visualization has the advantage of showing intuitively what information a neural net relies on to make a specific decision. For instance, in figure 2 [38], for the results in layer 5, row 1, col 2, it is noticeable that when just examining the image patches of layer 5 it seems that they have nothing in common. But after looking at its feature visualization we realize that it detects the grass in the background. However, the biggest disadvantage of this approach is how to measure the interpretability quantitatively (to what extent an interpretation is better than another one), which is also a common problem for other visualization techniques.

To overcome this disadvantage, we Apply t-SNE on the same dataset and use neighborhood hit (NH) [26] to measure the projection quality, which generates values to quantitatively interpret a neural net. For a specific point p, we select its k nearest neighbours and calculate its NH: NH_p as the ratio of the number of points belonging to the same class c as p. The NH for all the points of the projection is the average of NH over all the points.

$$NH_p = \left.\frac{N_c}{N}\right|_k \tag{2}$$

$$NH = \frac{\sum NH_p}{N} \tag{3}$$

Our proposed approach is shown in Fig. 1. In this figure, we present our method in the case when we apply transfer learning based on Alexnet. The ImageNet data is first used to train the Alexnet and then the pretrained model is applied on the CIFAR-10 dataset. The blue blocks in Fig. 1 represent the portions that the two networks share (before transfer and after transfer). The green blocks are the fully connected layers, which are different for the two networks because the ImageNet dataset has 1000 classes while the CIFAR-10 dataset has only 10 classes. After the transfer learning process, both feature visualization (we use Deconvnet) and data visualization (we use t-SNE) are applied on the hidden layers of the transferred neural nets. Then we compare their results. When refining a neural net, both of the visualizations could be combined for better detection.

The pseudo code of our algorithm is shown in Algorithm 1.

In this algorithm, for the hidden activation values h_i at each hidden layer, we first compute the conditional probability $p_{j|i}$ which is then applied to calculate the joint probability p_{ij}. The low dimensional representation is then calculated iteratively within T iterations employing the derived gradient in equation (4).

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|x_i - x_k\|^2/2\sigma_i^2)} \tag{4}$$

Here σ_i is the variance of the Gaussian distribution. Meanwhile, each m-dimensional hidden node x_i will have a corresponding d-dimensional counterpart y_i. The similarity between y_i and y_j could also be modeled as a probability $q_{j|i}$. We can express $q_{j|i}$ as follows:

$$q_{j|i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k\neq i}\exp(-\|y_i - y_k\|^2)} \tag{5}$$

[1] The visualization results are here: http://ai.stanford.edu/ quocle/TCNNweb/index.html

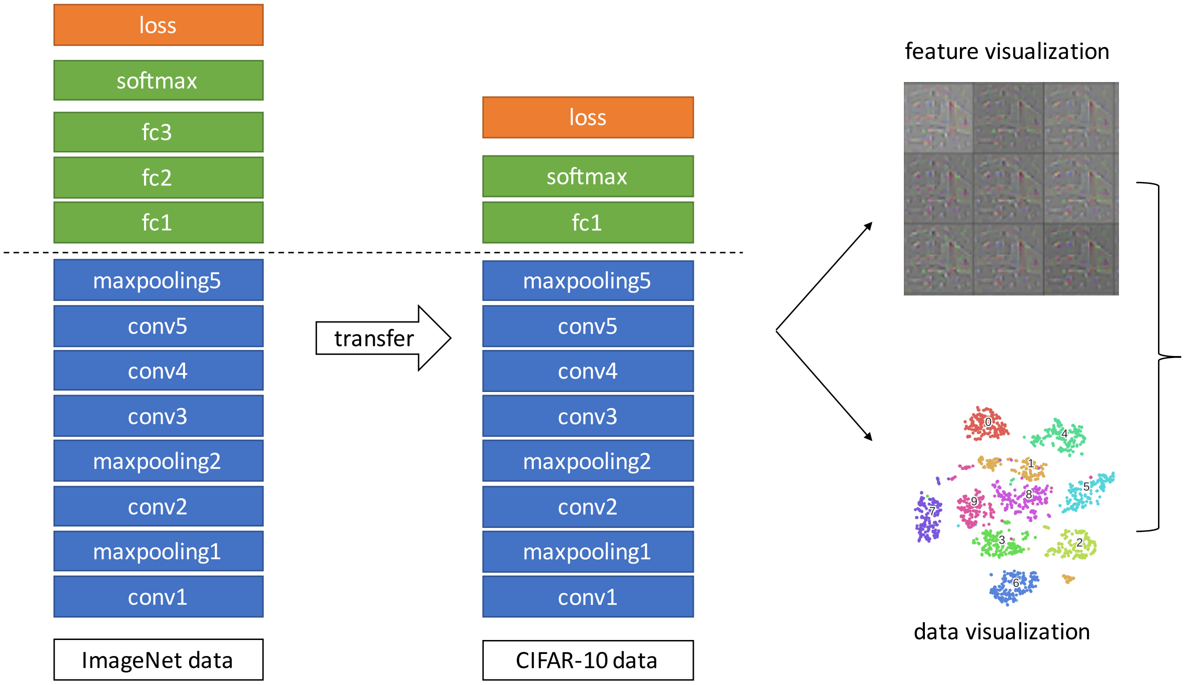


Figure 1: Architecture for refining deep convnets.

Algorithm 1

```
Input: training dataset $C_{tr}$; test dataset $C_{te}$; dimension reduction parameters: perplexity $perp$, iterations $T$, learning rate $\eta$, momentum $\alpha(t)$; pre-trained neural net model: $M$.
$C_{tr} \xrightarrow{resize} (C_{tr})^r$
$M_n = M((C_{tr})^r)$
$C_{te} \xrightarrow{resize} (C_{te})^r$
$H = M_n((C_{te})^r)$
for hidden activations $h_i = h_1, \cdots, h_H$ do
    compute $p_{j|i}$ with $perp$
    set $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2n}$
    Initialize $Y^{(0)}$
    for $t = 1$ to $T$ do
        compute $q_{ij}$ in $d$-dimensional space
        compute gradient $\frac{dL}{dy}$
        set $Y^{(t)} = Y^{(t-1)} + \eta\frac{dL}{dy} + \alpha(t)(Y^{(t-1)} - Y^{(t-2)})$
    end for
    compute $NH$ for $Y$
end for
Output: $\{NH_1, NH_2, \cdots, NH_H\}$
```

3 EXPERIMENTS

We apply our approach on two deep neural nets: Alexnet [13] and VGG19 [32]. Both of them are originally trained on the ImageNet dataset [29]. We then apply the pre-trained models on the CIFAR-10 dataset and fine tune the transferred models. It's noticeable that the image size for the ImageNet dataset is 224 × 224 × 3 while that is 32 × 32 × 3 for the CIFAR-10 dataset. Hence, we resized the image size of CIFAR-10 dataset to 224 × 224 × 3 to fit the data to the pre-trained neural nets. The details for the CIFAR-10 dataset is shown in Table 1 and the parameters used for fine tuning the pre-trained models on CIFAR-10 is displayed in Table 2. The test accuracy for Alexnet on CIFAR-10 is 79% and 91% for VGG19. The code for our experiments can be downloaded from the following GitHub repository.[2]

Table 1: CIFAR-10 Dataset

Dataset Details			
#Features	#Train	#Test	Classes
$32 \times 32 \times 3$	50,000	10,000	10

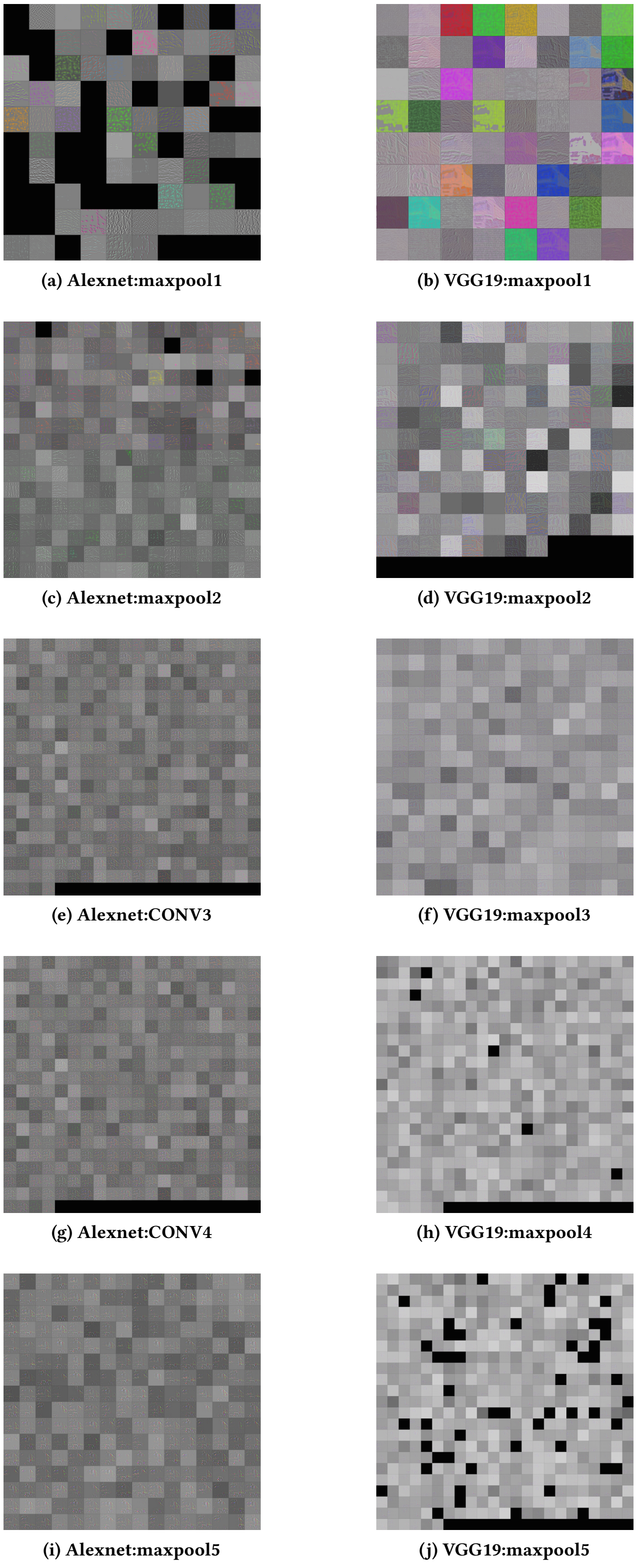
Deconvnet results. To do Deconvnet visualization, for illustration, we randomly pick a test instance showing a truck. The original image of this test instance is shown in Fig. 2. We then use the Deconvnet technique to generate the visualizations of the input image reconstructed from each of the feature maps on the specified hidden layers for both neural nets. Fig. 3 shows a comparison of the results. Within each picture, each block represents for the

[2] https://github.com/jadecranberry/Quantified-Data-Visualization-QDV/tree/main

Table 2: Parameter settings for fine tuning

Alexnet & VGG19		
learning rate	batch size	#epochs
0.00001	16	10

Figure 2: A test instance for Deconvnet visualization.



(a) Alexnet:maxpool1

(b) VGG19:maxpool1

(c) Alexnet:maxpool2

(d) VGG19:maxpool2

(e) Alexnet:CONV3

(f) VGG19:maxpool3

(g) Alexnet:CONV4

(h) VGG19:maxpool4

(i) Alexnet:maxpool5

(j) VGG19:maxpool5

Figure 3: Deconvnet results

visualization generated by each feature map within a specific hidden layer. For instance, the first maxpooling layer of Alexnet has 96 feature maps and hence there are 96 visualization blocks. Please note that these figures need to be magnified to observe differences.

We can infer from the figures that for each neural net, higher hidden layers would detect more specific visualizaitons: revealing more obvious features contributing to a truck. But it is difficult to deduce whether the visualizations of vgg19 are truely better than that of Alexnet on the hidden layers when just inspecting the figures, especially for the hidden layers after the 3rd hidden layer. Therefore, in the second step, we apply t-SNE to quantify the visualization results.

t-SNE results. In order to obtain quantitative visualization results, we randomly subsampled 1000 test instances from the original 10,000 test instances. Then we extracted the values of the hidden activations for these test instances corresponding to each of the hidden layers. Subsequently, we normalized the data and apply t-SNE on these data. We use NH to quantity the quality of projections. The results for both neural nets are shown in Table 3 and Fig. 4. It is noticeable that the NH values are comparable in the first two hidden layers and VGG19 has higher NH values than Alexnet for the rest of the hidden layers.

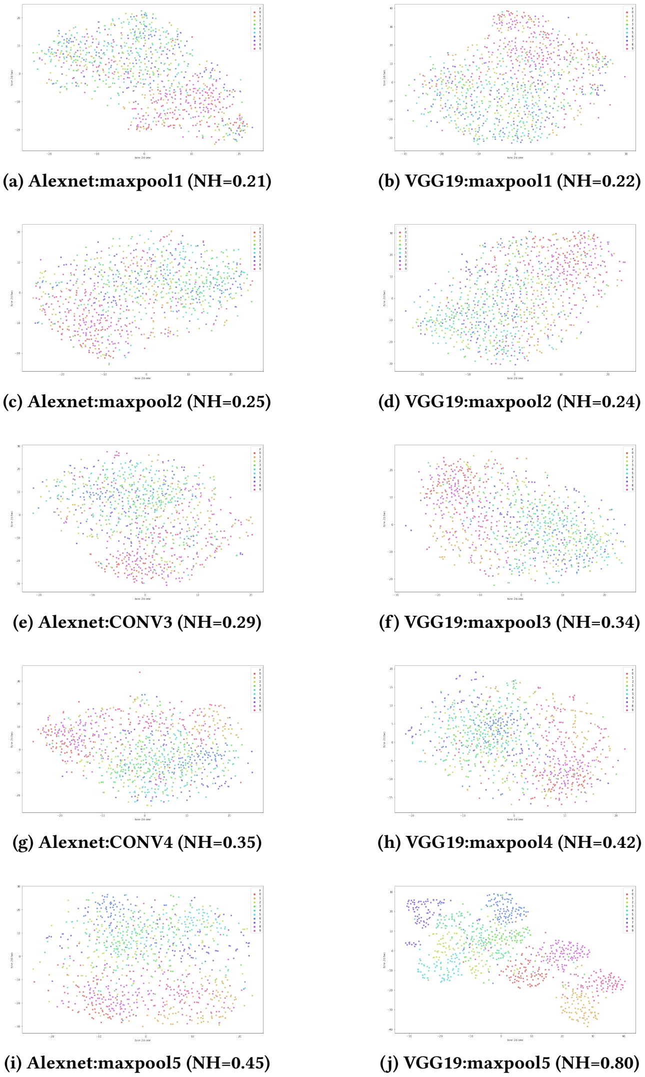
It should be noted that a better neural net structure doesn’t necessarily mean that at every hidden layer its NH value should be higher. In this case, although the NH values are comparable for these two neural net at the first and second layers, VGG19 gradually surpass Alexnet on the higher layers and especially on the last layer. This conclusion also agrees with the results in [38]. They changed the architecture in the first hidden layer, but we can’t observe much visual difference in this layer. However, the difference seems to be clearer in the second hidden layer. We consider this as a latent effect in the architecture change.

Discussion. The quantitative results on t-SNE visualization indicate that VGG19 indeed has a better structure than Alexnet, which provide stronger proof than just using the Deconvnet approach. By applying transfer learning, we can also avoid the performance

Table 3: NH for t-SNE at each hidden layer

hidden layers	Alexnet	VGG19
1st	0.21	0.22
2nd	0.25	0.24
3rd	0.29	0.34
4th	0.35	0.42
5th	0.45	0.80

loss of distilling a complex model (teacher) into a simpler one (student) [16]. Also, transfer learning on a simpler dataset (CIFAR-10) enables more clearer t-SNE visualizations than those on the ImageNet data [11].



(a) Alexnet:maxpool1 (NH=0.21)

(b) VGG19:maxpool1 (NH=0.22)

(c) Alexnet:maxpool2 (NH=0.25)

(d) VGG19:maxpool2 (NH=0.24)

(e) Alexnet:CONV3 (NH=0.29)

(f) VGG19:maxpool3 (NH=0.34)

(g) Alexnet:CONV4 (NH=0.35)

(h) VGG19:maxpool4 (NH=0.42)

(i) Alexnet:maxpool5 (NH=0.45)

(j) VGG19:maxpool5 (NH=0.80)

Figure 4: t-SNE results

4 CONCLUSIONS

In this paper, we attempt to diagnose and refine neural nets more convincingly. The experiments are implemented on two neural nets: the Alexnet and the VGG19. Using transfer learning, we are able to use the pretrained neural nets on the CIFAR-10 dataset. Employing the quantitative results on t-SNE visualization, we are able to reveal quantitatively why VGG19 has higher accuracy than Alexnet on the same dataset. This conclusion is more convincing than that in [38] when they just use the visualization of Deconvnet to justify their selection of filter size and stride of convolutional layer. Also applying transfer learning enables us to interpret the neural nets directly on a simper dataset, without distilling a complex model into a shallow model as in knowledge distillation. This is more advantegeous of maintaining the accuracy of the complex model while still enables easier interpretation on a smaller dataset.

REFERENCES

[1] Pietro Berkes and Laurenz Wiskott. 2006. On the Analysis and Interpretation of Inhomogeneous Quadratic Forms as Receptive Fields. *Neural Computation* 18, 8 (2006), 1868–1895.

[2] Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. 2015. Distilling Knowledge from Deep Networks with Applications to Healthcare Domain. *arXiv preprint arXiv:1512.03542* (2015).

[3] Trevor F Cox and Michael AA Cox. 2000. *Multidimensional Scaling.* Chapman and Hall/CRC.

[4] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2009. Visualizing Higher-Layer Features of a Deep Network. *University of Montreal* 1341, 3 (2009), 1.

[5] Nicholas Frosst and Geoffrey Hinton. 2017. Distilling a Neural Network into a Soft Decision Tree. *arXiv preprint arXiv:1711.09784* (2017).

[6] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2019. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys (CSUR)* 51, 5 (2019), 93.

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531* (2015).

[8] Geoffrey E Hinton and Sam T Roweis. 2003. Stochastic Neighbor Embedding. In *Advances in Neural Information Processing Systems.* 857–864.

[9] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 5786 (2006), 504–507.

[10] Harold Hotelling. 1933. Analysis of a Complex of Statistical Variables into Principal Components. *Journal of Educational Psychology* 24, 6 (1933), 417.

[11] Andrej Karpathy. 2014. t-SNE Visualization of CNN Codes. http://cs.stanford.edu/people/karpathy/cnnembed/.

[12] Pang Wei Koh and Percy Liang. 2017. Understanding Black-Box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning.* JMLR. org, 1885–1894.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems.* 1097–1105.

[14] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2018. Interpretable & Explorable Approximations of Black Box Models. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12, 5 (2018), 56.

[15] X. Li, H. Xiong, X. Li, et al. 2022. Interpretable Deep Learning: Interpretation, Interpretability, Trustworthiness, and Beyond. *Knowledge and Information Systems* 64, 12 (2022), 3197–3234. https://doi.org/10.1007/s10115-022-01756-8

[16] Xuan Liu, Xiaoguang Wang, and Stan Matwin. 2018. Improving the Interpretability of Deep Neural Networks with Knowledge Distillation. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW).* IEEE, 905–912.

[17] Xuan Liu, Xiaoguang Wang, and Stan Matwin. 2018. Interpretable Deep Convolutional Neural Networks via Meta-Learning. In *2018 International Joint Conference on Neural Networks (IJCNN).* IEEE, 1–9.

[18] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30,* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4765–4774.

[19] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems.* 4765–4774.

[20] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data Using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[21] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2018. Methods for Interpreting and Understanding Deep Neural Networks. *ACM Transactions on Intelligent Systems and Technology (TIST)* 9, 3 (2018), 1–15.

[22] Nature.com. 2023. Deep Neural Networks Display Impressive Performance but Suffer from Limited Interpretability. https://www.nature.com/articles/s41540-023-00310-8. Accessed: 2024-05-19.

[23] Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang W Koh, Quoc V Le, and Andrew Y Ng. 2010. Tiled Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*. 1279–1287.

[24] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2016. Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned by Each Neuron in Deep Neural Networks. *arXiv preprint arXiv:1602.03616* (2016).

[25] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. 2018. The Building Blocks of Interpretability. *Distill* 3, 3 (2018), e10.

[26] Fernando V Paulovich, Luis G Nonato, Rosane Minghim, and Haim Levkowitz. 2008. Least Square Projection: A Fast High-Precision Multidimensional Projection Technique and Its Application to Document Mapping. *IEEE Transactions on Visualization and Computer Graphics* 14, 3 (2008), 564–575.

[27] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. *ACM SIGKDD Explorations Newsletter* 18, 1 (2016), 1135–1144.

[28] Andrew S. Ross and Finale Doshi-Velez. 2018. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[30] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2017. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Transactions on Neural Networks and Learning Systems* 28, 11 (2017), 2660–2673.

[31] Ravid Shwartz-Ziv and Naftali Tishby. 2017. Opening the Black Box of Deep Neural Networks via Information. *arXiv preprint arXiv:1703.00810* (2017).

[32] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556* (2014).

[33] Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo. 2019. Learning Global Additive Explanations for Neural Nets Using Model Distillation. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 4 (2019), 1–23.

[34] Jayaraman J Thiagarajan, Bhavya Kailkhura, Prasanna Sattigeri, and Karthikeyan Natesan Ramamurthy. 2016. TreeView: Peeking into Deep Neural Networks via Feature-Space Partitioning. *arXiv preprint arXiv:1611.07429* (2016).

[35] Warren S Torgerson. 1952. Multidimensional Scaling: I. Theory and Method. *Psychometrika* 17, 4 (1952), 401–419.

[36] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. 2009. Dimensionality Reduction: A Comparative. *Journal of Mach Learn Res* 10, 66-71 (2009), 13.

[37] Tom Zahavy, Nir Ben-Zrihem, and Shie Mannor. 2016. Graying the Black Box: Understanding DQNs. In *International Conference on Machine Learning*. 1899–1908.

[38] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer Vision*. 818–833.

[39] Matthew D Zeiler, Graham W Taylor, Rob Fergus, et al. 2011. Adaptive Deconvolutional Networks for Mid and High Level Feature Learning. In *ICCV*, Vol. 1. 6.

[40] Quan-shi Zhang and Song-Chun Zhu. 2018. Visual Interpretability for Deep Learning: A Survey. *Frontiers of Information Technology & Electronic Engineering* 19, 1 (2018), 27–39.