Shedding Light on Deep Neural Networks:
Illumination Recognition, Fine-tuning, and Sequence Effects.

Eric James McDermott, Gerrit Alexander Ecke, Hanspeter A. Mallot

Abstract

Artificial Neural Networks can be thought of as complex pattern learning machines that pass input through a series of layers which perform different operations in order to produce a categorical-type output. Through supervised learning, these networks can come to be very accurate at certain categorization tasks. The current study consists of a novel dataset compiled from existing illumination databases of faces, objects, and textures. Each image was then binned or rendered in one of nine uniquely defined illumination categories. Data augmentation through mirroring, colorspace manipulations, and resizing allowed for a large sample set of over 85,000 images. In order to study the transferability of learning in neural networks, we compared learning rates and accuracy through two distinct approaches to the task of classifying illumination directions. First, we examined the training method through 'fine-tuning', whereby the pre-trained weights of the 2014 ImageNet winner GoogLeNet were utilized in comparison to a newly initialized network. Second, hierarchical learning, or sequence effects of input types were examined. Ultimately, the best combinations produced results consisting of a top-5 categorization accuracy of 100% and top-1 categorization accuracy of ~96% while providing further evidence toward 'fine-tuning' being a catalyst for the learning rate of deep neural networks. Furthermore, interesting dissociations were found regarding sequencing of input.

Introduction

In the past years the ability for neural networks to classify and detect objects has reached a new level [1]. This increase in ability largely resides in the creation of more efficient network architectures, such as GoogLeNet [2], and AlexNet [3]. The primary task for these networks is to take an input typically in the form of an image, pass it through a series of convolutional, pooling, thresholding, and fully-connected layers. The convolutional layer serves as a filter, whereby the image is convolved into another representation according to certain weighting factors. A single image will be viewed through many of these filters through a pre-specified kernel, allowing different types of extraction to occur. The pooling layers serve as a way to reduce the overall spatial dimension of the inputs, as well as introduce some noise into the system by discarding lower activations. Whereas the thresholding layers pass on input above a certain threshold, and inhibit the input below the threshold. All together, this information is funneled into a series of fully-connected layers which in the end categorize the input into a particular classification. This is a form of "supervised" learning. In general, upon providing the categorization, the network then acquires an error signal dependent on the prior categorization. This error signal is then back-propagated throughout the aforementioned layers, slightly altering the weighting parameters. This method attempts to reduce the *loss* -or the error- of the network. Over time the network will establish strong weighting parameters to extract information at each layer in order to best produce the desired output. Interestingly, these weighting parameters have been shown to not only be relevant to the task at hand, but also generalized to improve learning rates on slightly different tasks [4, 5]. This concept is further explored in the current project, whereby dimensionality is reduced and completely different categorizational goals are given to neural network. In this case, the task was to identify from where illumination was coming from in the image. Continuing with this idea of *fine-tuning,* we had the idea that since there is evidence that building upon prior structures is beneficial for the learning rate of neural networks, perhaps there is some relevance in the order of category learning within a single task. For example, would a dataset consisting of faces, objects, and textures be better learned if first lower level features inherent in textures are presented, and then objects and faces? Or would it learn better distinguishing large scale illumination shifts such as those apparent in faces and objects? The study at hand set out to investigate such questions, as well as the extent of benefits found through network *fine-tuning*.

Methods

To carry out our investigation we utilized the Caffe framework found in [5], written in C++ and used through the terminal interface. The pre-trained weights involved in the fine-tuning aspect were those of the 2014 ImageNet winner, GoogLeNet [2]. Our database was compiled from parts of three existing illumination databases, the Yale Face Database [6], the Amsterdam Library of Object Images [7], and the KTH-TIPS2 database [8]. In order to acquire a large enough amount of input data, the aforementioned databases were first rendered as 256x256 pixel .jpg images and then mirrored over the vertical axis, as well as the horizontal axis (with the exception of the facial database). Images were also rotated +/- 45 degrees and then centrally cropped. After this process, the images were binned in 9 unique illumination conditions corresponding to center illumination, left illumination, top-left illumination, top illumination, top-right illumination, right illumination, bottom-right illumination, bottom illumination, and bottom-left illumination, as seen in Figure 1. When illumination fell between two categories, the mean angle of degree between the categories was used as a boundary. The images were then converted into both 3-channel RGB and 3-channel greyscale images to increase the possible depth of the input layer extraction. Altogether, this procedure rendered over 60,000 images. Furthermore, the network itself was instructed to take a 228x228 pixel crop of an image for processing during each iteration, increasing the variability of the samples. Data analysis was done with MATLAB.

The training procedure for using pre-trained network weights consisted of obtaining a saved *solverstate (the updated weights and learning state)* from GoogLeNet after the network learned on the ImageNet database. From this solverstate, the final layers outputs were converged onto the 9 categories specified in the current project, and then training was resumed using the aforementioned database. The sequence effects consisted of first training the network either on a database consisting of the face and object images or one consisting of the texture images. After ~550,000 iterations, the lacking database was compiled with the other and the network continued learning using the weights from the previous solverstate. In general, the neural networks were allowed to run until error rate decreases were plateauing, ultimately limited by the resources and time of the researchers.

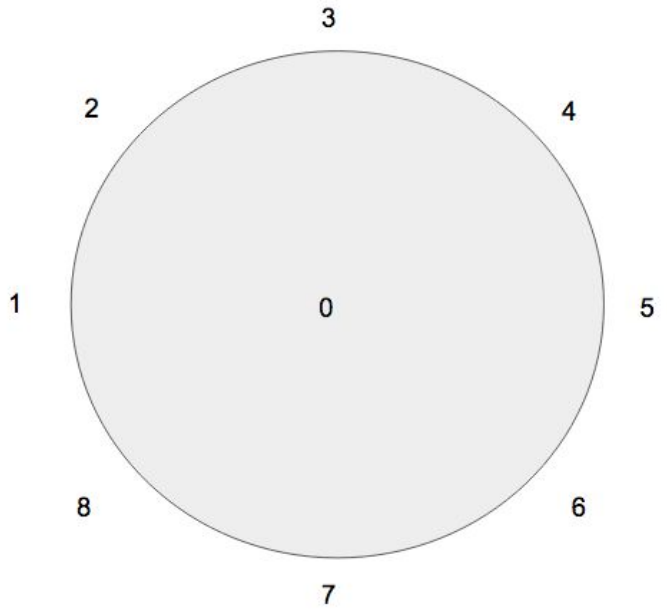


Figure 1: The binned illumination categories came from the 9 directions seen above.

Results

One major finding of the study is that utilizing a network with pre-trained weights significantly sped up the learning rate, as can be seen through Figure 2. Changes were also seen regarding the differences in learning rate when networks were trained with different hierarchical learning patterns, such as first training faces and objects and then adding in the textures at a later iteration (~3% change [0.03 error rate] over 1,250,000 iterations in the new network and ~3% change over 950,000 iteration in the pre-trained network), as is depicted in both the pre-trained and new networks on Figure 3 and 4. It is apparent that the network learned better when first given faces and objects, and then textures. This pattern was seen in both the newly initialized network and the pre-trained network. However, even when attempting different sequence effects, the final accuracy never outperformed the pre-trained network when given all three feature sets simultaneously. With this network, the top-5 categorization accuracy was at 100%. In addition, the pre-trained network achieved a 97% top-1 accuracy in 555,000 iterations in comparison to the 93% top-1 accuracy the newly initialized network achieved in 1,250,000 iterations when trained on all features.

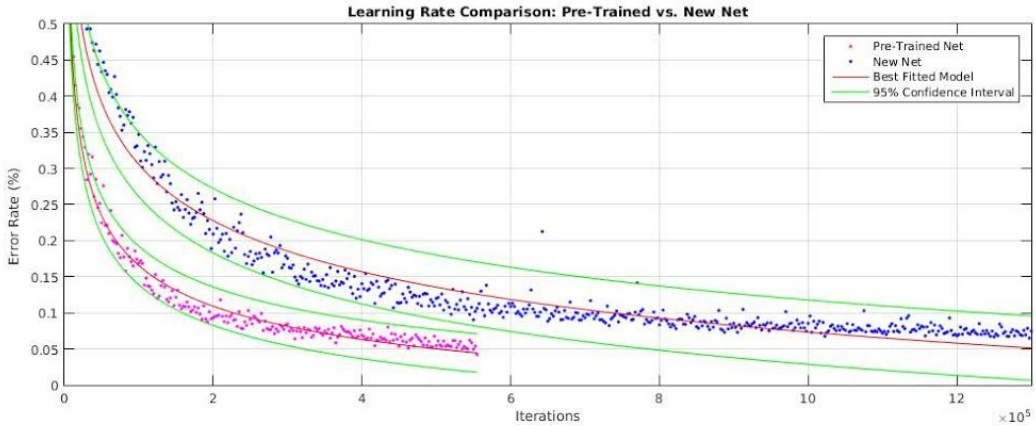


Figure 2: The pre-trained network (shown in magenta) significantly outperforms the newly implemented network (shown in blue) as a function of error rate reduction. 95% confidence intervals were constructed around the best fit.

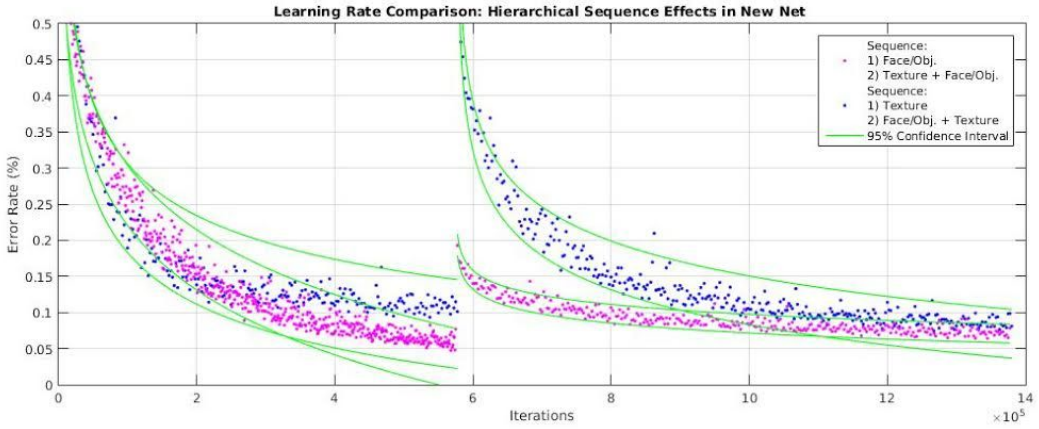


Figure 3: The hierarchical learning paradigm within the pre-trained network shows an effect in error rate of around 3% increased accuracy with training faces and objects first and then training textures once the first stage is completed, as can be seen by the lower final state of the magenta points versus the blue points.

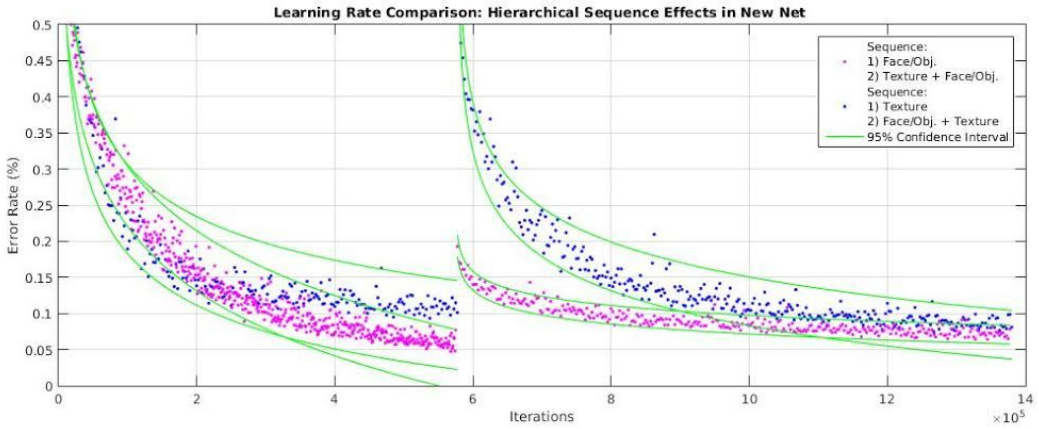


Figure 4: The hierarchical learning paradigm within the newly initialized network shows an effect in error rate of around 3% [0.03 error rate] increased accuracy with training faces and objects first and then training textures once the first stage is completed, as can be seen by the lower final state of the magenta points versus the blue points.

Discussion

We believe that these results significantly support the use of pre-trained neural networks when training novel categories for object recognition. We were able to use novel inputs that not only were different that a pre-trained networks objective classifications, but asked the network to ignore its prior classifications and concentrate on a whole new task of categorizing illumination direction. The network was able to do this, and it was able to do it more efficiently and with greater accuracy. Furthermore, the use of sequencing and hierarchical learning paradigms may have some role in the future, but this will have to be explored more in depth in future experiments. Importantly, we must ensure that all input categories are of equal proportion (i.e.; faces, objects, textures). We must also allow the network to run until exhaustion, that is, until it reaches a clear plateau. It will be interesting to see if the hierarchical learning paradigm can produce an overall lower error rate, regardless of the iterations it takes to get to that point. We should also take measures to determine the spatial frequency inherent in each image category. The texture dataset was also subjectively much harder to determine the direction of illumination, as seen in Figure 5, and this could be another reason why the benefit of hierarchical learning was not fully realized. Interestingly, when utilizing deconvolutional networks to visualize the networks relevant activity, inspired from Zeiler and Fergus [9], we find that in contrast to the finely detailed representations the later layers showed in a network such as GoogLeNet trained on ImageNet (Figure 6), in our network a shift there appears to be a shift from representing objects to representing a light source (Figure 7). This can be thought of a visual confirmation that our network has now learned that the important feature is illumination direction, as would be required in order to properly categorize the images. In conclusion, it appears that even when a network is tasked with a completely different set of relevant features, the rate of learning can be improved by providing it with pre-trained weights of another feature detecting network. Future directions may include using a variety of pre-trained weights from different networks parameters, as well as compiling larger databases of illumination directions to further analyze hierarchical learning. Exploring the idea of hierarchical learning also is appealing, not only because we find slight evidence of it, but also when looking at the concept of applying pre-trained networks from a step back we see it is a form of hierarchical learning in itself.

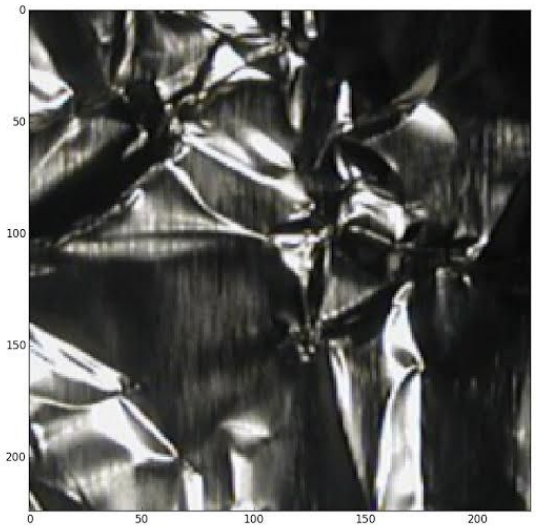


Figure 5: An example image from the texture database.

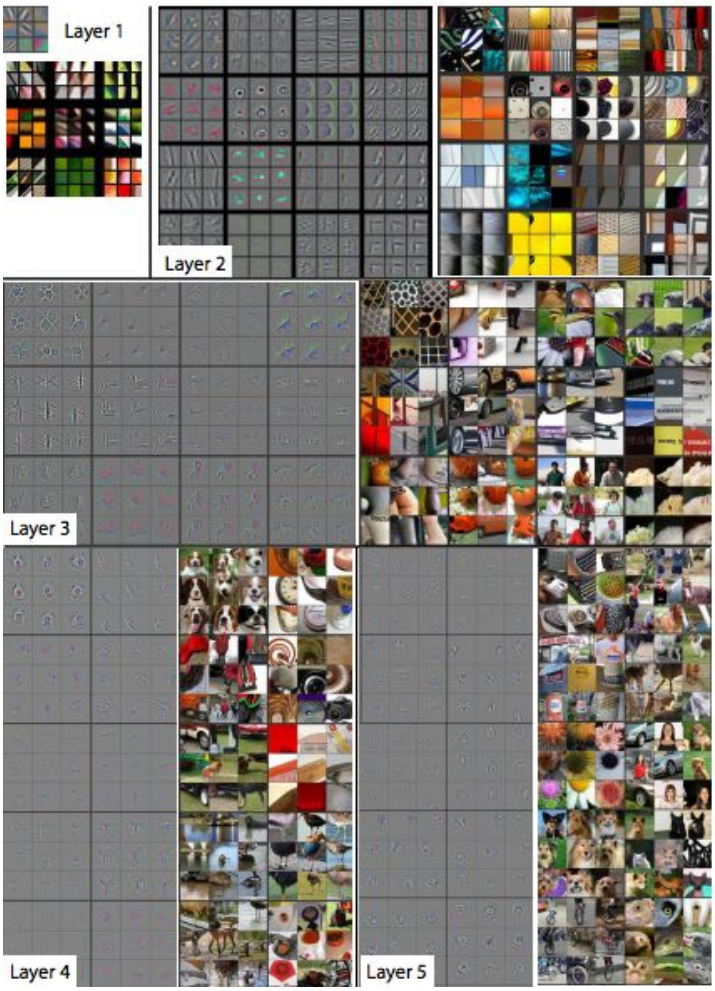


Figure 6: Example deconvolutions shown for various images in the progression of layers.
(Taken from [9])

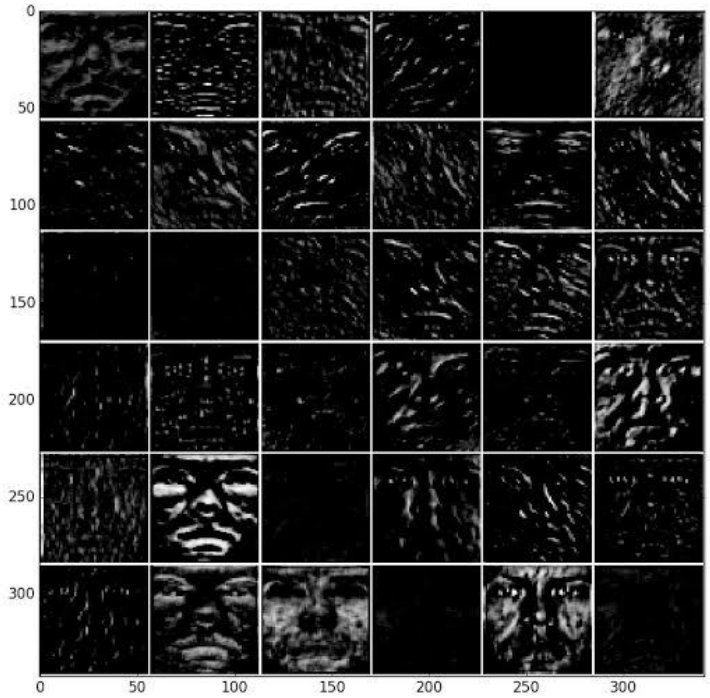
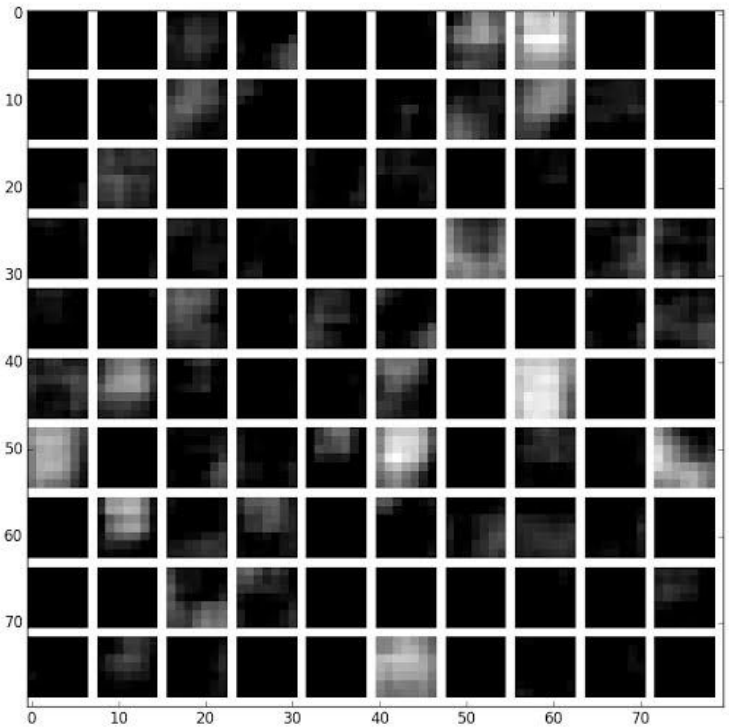


Figure 7: The image (below) passed through a series of convolutional filters produces images seen on the left, then a fully connected layer just prior to the final output layer represents what appears to be large depictions of light sources (bottom left).



References

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Comput., 1(4):541–551, Dec. 1989.

[2] Christian Szegedy, Wei Liu , Yangqing Jia , Pierre Sermanet, Scott Reed , Dragomir Anguelov , Dumitru Erhan , Vincent Vanhoucke , Andrew Rabinovich. Going Deeper with Convulitions. CVPR., 2015.

[3] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25, pages 1106–1114, 2012.

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In ICML, 2014.

[5] Yangqing Jia , Evan Shelhamer , Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. ARVIX. 2014.

[6] Kuang-Chih Lee, Jeffrey Ho, and David Kriegman. Acquiring Linear Subspaces for Face Recognition under Variable Lighting, PAMI, May, 2005

[7] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, The Amsterdam library of object images, Int. J. Comput. Vision, 61(1), 103-112, January, 2005

[8] P. Mallikarjuna, Alireza Tavakoli Targhi, Mario Fritz, Eric Hayman, Barbara Caputo and Jan-Olof Eklundh. The KTH-TIPS2 database, 2006

[9] M.D. Zeiler, R. Fergus. Visualizing and Understanding Convolutional Networks. ECCV 2014, Arxiv 1311.2901. Nov 28, 2013.