ON THE GENERALITY OF NEURAL IMAGE FEATURES

Ragav Venkatesan, Vijetha Gatupalli, Baoxin Li

School of Computing Informatics and Decision Systems Engineering,
Arizona State University, Tempe, AZ, USA.

ABSTRACT

Often the filters learned by Convolutional Neural Networks (CNNs) from different image datasets appear similar. This similarity of filters is often exploited for the purposes of transfer learning. This is also being used as an initialization technique for different tasks in the same dataset or for the same task in similar datasets. Off-the-shelf CNN features have capitalized on this idea to promote their networks as *best transferable* and *most general* and are used in a cavalier manner in day-to-day computer vision tasks.

While the filters learned by these CNNs are related to the *atomic structures* of the images from which they are learnt, all datasets learn similar looking low-level filters. With the understanding that a dataset that contains many such *atomic structures* learn general filters and are therefore useful to initialize other networks with, we propose a way to analyse and quantify generality. We applied this metric on several popular character recognition, natural image and a medical image dataset, and arrive at some interesting conclusions. On further experimentation we also discovered that particular classes in a dataset themselves are more general than others.

1. INTRODUCTION

Neural networks, particularly CNNs have broken all records recently in the computer vision research area. Large networks are often trained with large number of data samples to achieve good accuracies [1, 2]. While using CNNs as feature detectors has long been studied (e.g.,[3]), recent years large networks trained with large-scale datasets (e.g.,[4]) have started being used as "off-the-shelf" tools for feature extraction. Still, some studies show that a few ($< 1\%$) nodes are all that are actively contributing to classification [5]. While it is reasonable to expect edge detectors and Gabor-like features in the lower-level filters and more sophisticated concepts at the higher levels, it is not clear as to why these filters adapt themselves in this manner. What is fairly clear though is that different datasets result in different sets of filters that are similar if the datasets are similar. It is only natural to ask, what role does the images themselves play in such filters being learnt and how they compare with filters learnt from another dataset. In this paper we take the view that the filters learnt by networks when trained using a particular dataset represent the detectors for some *atomic structure* in the data itself. Each layer is a mapping form the previous layer to the next layer that is constructed using combinations of these atomic structures in the first layer in order to minimize a cost.

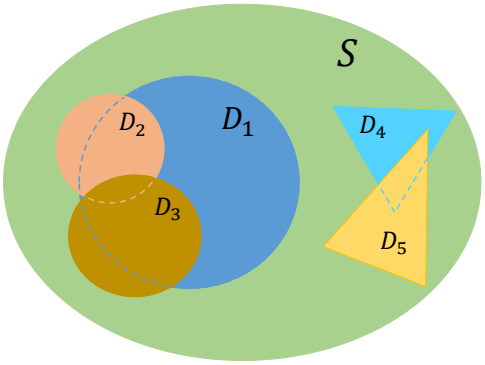

Fig. 1. Thought experiment to describe the dataset generality. S is the space of all possible atomic structures, $D_1 - D_5$ are the atomic structures present in respective datasets.

Let us first define *atomic structures* to be the forms that CNN filters take by virtue of the entropy of the dataset. At the first layer of a CNN, these might be the edge and blob detectors. Consider the following thought experiment: Let's assume that all possible atomic structures reside in an universe S. Suppose we have a set of three datasets $D = \{D_1, D_2, D_3\}$ and $D \in S$. Consider the system in figure 1. One would now recognize that D_1 is a more general dataset with respect to D_2 and D_3. It is so because, while D_1 *contains* most of the atomic structures of D_2 and D_3, the latter does not contain as many atomic structures of D_1. While this analysis is simplified for one layer, in typical CNNs, co-adaptation plays a major role in the learning of these atomic structures. Therefore, generality as defined by the overlap of areas in a layer-wise Venn diagram is impractical to obtain.

In this paper we postulate that, the generalization performances of CNNs on one dataset re-trained on a network initialized by the weights of another network trained using another dataset, could be used to derive generality between the said datasets. We call this process of pre-trained initialization

as *prejudicing*. By prejudicing on the first dataset, we froze[1] and unfroze layers and retrained the networks on the second dataset. When the prejudicing dataset is more general than the re-train dataset, the classifier generalizes better.

We developed a generality metric by comparing the gain in performances of networks of various obstination. Using a generality such as the one proposed, it becomes clear as to what kind of datasets are to be used to prejudice CNNs with, during transfer learning. We also discovered that samples with particular labels within a dataset alone are general enough that if we begin by training the network on only those and then moved on to the rest of the classes, we were able to learn the rest of the dataset with considerably less training samples while achieving comparable generalization performances. This study led us to two major research insights: 1.If one has very few data to learn from, which other dataset is better to prejudice the network with? 2. Among the various classes during the training procedure, if we prejudice with a certain *general* set of classes first and then move on to others later, generalization to all classes, even for those with few samples is better.

The rest of the paper is organized as follows: section 2 discusses related works, section 3 presents the design of our experiments, section 4 shows some results on the core-experiment and section 5 provides concluding remarks.

2. RELATED WORK

One prior art closely related to this article is [6]. In that article, the authors considered two tasks A and B that were essentially 500 classes each from the Imagenet dataset [7]. They experimented by obstination and prejudice, the specificity of each layer and their contributions to the overall performance. They also showed that networks working on similar tasks had a high memorability and that co-adaptation of layers increased the generalization performance. While this analysis is interesting, it was performed on only one dataset: Imagenet. By design, the networks were forced to learn very general filters, so as to be best transferable. Also, the paper analysed the transferability of the feature extractors from the perspective of the networks in terms of their fall in generalization performance. This analysis was not catered to the dataset's perspective, which is that the filters learned are a property of the dataset being trained on.

Another closely related work is the work by Hinton et al, on the transfering of softmax layers [8] . Here the authors suggest that among the various classes in a dataset, there exists some amount of generalization knowledge that could be transferred. They showed that the network learns the relationship between the classes even though not explicitly trained [8]. Although the author retrains an entire network that is randomly initialized using the softmax outputs from a trained network and uses this as prejudice, no information is actually being transferred in terms of actual filters. Ergo, this work, while interesting, also doesn't help in understanding generality of the data itself in a more direct manner. Some of the claims made by this article though were indirectly and independently verified by us through our generality results. The basic claim of their work is that among only a handful of classes, there is enough knowledge to generalize to other classes. Unless there exists some generality between classes, training on particular classes will not have been representational enough for the other classes. We directly verify this by showing that some classes alone have a high generalization to the rest of the dataset and make a similar conclusion from an entirely independent direction of research.

[1]Obstinate layer or freezing implies that the weights were not changed during backprop. The layer remains prejudiced.

3. DESIGN OF EXPERIMENTS

We designed these experiments across three broad categories of datasets:1. Character datasets that included MNIST [9], MNIST-rotated [10], MNIST-random-background [10], MNIST-rotated-background [10], Google street view house numbers [11], Char 74k English [12] and Char 74k Kannada [12] 2. Natural image datasets that includes Cifar 10 and Caltech 101 [13, 14] and 3. Natural images against medical images that included in addition to Caltech 101 a Colonoscopy video quality dataset. We leave it to the reader to find for themselves details about the datasets from the original articles. Details of the datasets used, network architecture and other logistical details are discussed in the supplementary. The experiments were designed using Theano, and the code is available here [15].

Among the various datasets used, it is natural to expect any network trained on MNIST to contain simpler filters than MNIST-rotated. This is because, while MNIST-rotated contains many structures from MNIST, due to the rotations, MNIST-rotated will contain additional structures that require the learning of more *complicated* filters. A network trained on MNIST-rotated on its first layers will be expected to additionally have filters for detecting sophisticated oriented edges than for MNIST. This would mean that prejudicing a network with MNIST to then re-train MNIST-rotated is much less helpful than vice versa. A network prejudiced with a general enough dataset is better to be retrained for it generalizes easily. A prejudice must come from a more general dataset if a prejudice transfers positive knowledge as shown in their generalization performances. We use this simple intuition to argue that MNIST-rotated is a more general dataset with respect to MNIST.

Our basic experiment was conducted between pairs of datasets D_i and D_j. Firstly, we train (prejudice) a randomly initialized network with dataset D_i. We call this network $n(D_i|r)$ or the base network (r implies random initialization). We then proceed to retrain $n(D_i|r)$ as per any of the setup shown in figure 2. $n_k(D_j|D_i)$ would imply that there

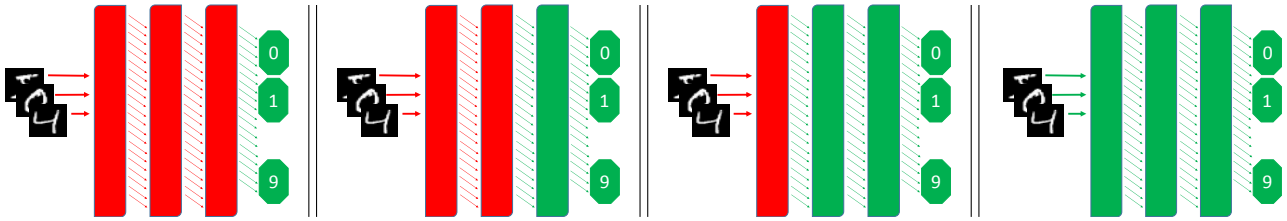

Fig. 2. Protocol of obstination: From left to right, all layers frozen, one, two and three layers unfrozen. Green represent unfrozen and red represent frozen. Note that the layers are always unfrozen from the end and that the softmax layer is always unfrozen and randomly initialized. This should be generalized similarly for more than three layers also.

are k degrees of freedom, or to be precise, k layers of filters that are allowed to learn by dataset D_j that is prejudiced by the filters of $n(D_i|r)$. $n_k(D_j|D_i)$ has $N-k$ obstinate layers that carries the prejudice of dataset D_i, where N is the total number of layers. Note that more degrees of freedom implies that the network is less obstinate to learn. Also note that these layers can be both convolutional or fully connected neural layers. Any idea expressed here can be extended to any type of parametrized layers. In fact while we perform operations such as batch normalizations, we even freeze and unfreeze the α and β of batch norm [16].

Suppose the generalization performance of $n(D_j|r)$ is $\Psi(D_j|r)$ and the generalization performance of $n_k(D_j|D_i)$ is $\Psi_k(D_j|D_i)$. Dataset generality of D_i with respect to D_j at the layer k is given by,

$$g_k(D_i, D_j) = \frac{\Psi_k(D_j|D_i)}{\Psi(D_j|r)} \tag{1}$$

This indicates the level of performance that is achieved by D_j using $N-k$ layers worth of prejudice from D_i and k layers worth of features from D_i combined with k layers of novel knowledge from D_j together. $g_k(D_i, D_j) > g_k(D_i, D_l)$ indicates that at k layers, D_i provides more general features to D_j than to D_l. Conversely, when initialized by $n(D_i|r)$, D_j has an advantage in learning than D_l. Note that, $g_k(D_i, D_i) \geq 1 \ \ \forall k$. $g_k(D_i, D_j)$ for $i \neq j$ might or might not be greater than 1.

D_i and D_j need not be entire datasets but can also be just disjoint class instances of the same dataset that is split in two. For instance, we divided the MNIST dataset into two parts. The first part contained the classes $[4, 5, 8]$, the rest were contained by the second part. We performed the generality experiments with MNIST$[4, 5, 8]$ as base. We re-trained this prejudiced network using the second part with the same experiment design as above. We repeated this experiment several times with decreasing number of training samples per-class in the retrain dataset of MNIST $[0, 1, 2, 3, 6, 7, 9]$. The testing set remained the same size. We created seven such datasets with $7p,\ p \in [1, 3, 5, 10, 20, 30, 50]$ samples each. We found that initializing a network that was trained on only a small sub-set of well-chosen classes can significantly improve generalization performance on all classes, even if trained with arbitrarily few samples, even at the extreme case of one-shot learning.

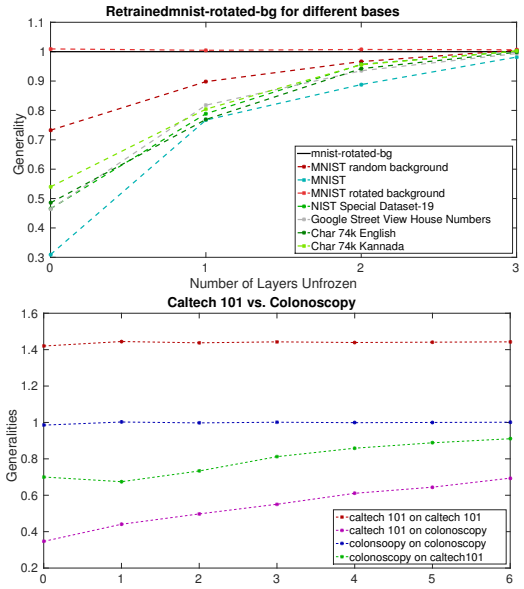

Fig. 3. Generalities. The dark line represents the accuracy of $n(D|r)$. Please zoom on a computer monitor for closer inspection. More plots for other combinations of datasets are included in the supplementary.

4. RESULTS AND OBSERVATIONS

Figure 3 shows the generalities of MNIST-rotated-bg and Kannada prejudiced by all other the character datasets. For reference each plot also shows the generalization performance of a randomly initialized base convolutional network. The following are some observations of interest:

While no dataset is qualitatively the most general, it is quite clear that *MNIST dataset is the most specific*. Rather, MNIST dataset is one that is generalized by all datasets very highly at all layers. Surprisingly, MNIST dataset actually gives better accuracy when prejudiced with other datasets,

p	base	$k=0$	$k=1$	$k=2$	$k=3$
1	Random	-	-	-	55.61
	MNIST[458]	73.07	73.91	76.37	77.52
3	Random	-	-	-	73.34
	MNIST[458]	83.61	87.2	85.7	87.6
5	Random	-	-	-	83.32
	MNIST[458]	90.98	92.98	92.6	92.07
10	Random	-	-	-	81.31
	MNIST[458]	91.55	93.71	93.82	95.08
20	Random	-	-	-	87.77
	MNIST[458]	95.52	95.52	97.07	96.78
30	Random	-	-	-	88.62
	MNIST[458]	96.5	97.34	97.35	97.45
50	Random	-	-	-	90.78
	MNIST[458]	96.38	97.40	97.71	97.38

Table 1. Sub-sample experiment and its generalization accuracies for different layers of freezing. The re-train network was MNIST$[0, 1, 2, 3, 6, 7, 9]$. For obvious reasons random initializations are trained only with all layers unfrozen, hence the missing values.

rather than when initialized with random, if all layers were allowed to learn. This is a strong indicator that *all datasets contain all atomic structures of MNIST*.

While initially one would have assumed that Kannada would be a general dataset, we observed the contrary. SVHN, Char74-English and Nist generalizes better to Kannada than even Kannada itself does. *English characters seem to be a more general set than Kananda.* While counter-intuitive, this result is immediately obvious when one pays close attention to the filters that are learnt and the dataset itself. Kannada is dominated by predominantly curved edges only, whereas even MNIST has a multitude of unique atomic structures.

For the intra-class experiment described above, table 1 shows the accuracies. From the table one can observe that even with one-sample per class, a 7-way classifier could achieve 22% more accuracy than a randomly initialized network. It is note worthy that the last row of table 1 still has 100 times less data than the full dataset and it already achieves close to state-of-the-art accuracy even when no layer is allowed to change. This is a remarkably strong indicator that the classes $[4, 5, 8]$ generalizes the entire dataset. We also observed that once initialized with a general enough subset of classes from within the same dataset, the generalities didn't vary among the layers like it did when we initialized with data from outside the mother dataset. We also observed that the more the data we used, more stable the generalities remained. Point of take away from this experiment is that if the classes are general enough, one may initialize the network with only those classes and then learn the rest of the dataset even with very small number of samples.

The colonoscopy dataset's labels identify if a image is deemed to be of a quality that is good enough so as to make a diagnosis on the pathology of that particular image. Most often the video quality in colonoscopy is affected because of saturation when too much light is thrown at a scene. The quality is also affected due to light reflection from bodily fluids that is also noticeable in the activations. Most of the filter colors are yellowish or blueish. On an colonoscopy video most often the video is also labelled poor quality when these colors are present, as these colors are often present mostly because of scattering and reflections. Having made these observations one would arrive at the obvious conclusion that neither dataset generalizes the other. This was indeed the result observed from figure 3. Although, Caltech 101 seem to generalize a bit better for even though it predominantly learns shapes, it learns some color features also.

From all these results and observations, we could summarize that one should prefer to initialize with a general dataset that might have a lot of variability or rather generality in data, when attempting to train with very few number of samples. Whenever possible one must initialize the network trained by a general dataset as this always boosts generalization performance. When there are biased datasets with large number of samples in some classes and fewer in others, one should train the most general classes first. Once the network is well-prejudiced one should start introducing the classes with fewer number of and less general samples, provided the general class is general enough.

5. CONCLUSIONS

In this paper, we used the performance of CNNs on a dataset when initialized with the filters from other datasets as a tool to measure generality. We proposed a generality metric using these generalization performances. We used the proposed metric to compare popular character recognition datasets and found some interesting patterns and generality assumptions that add to the knowledge-base of these datasets. In particular, we noticed that MNIST data is one of the most specific dataset. We also found that Char74k Kannada is less general than English datasets. We also calculated generality on class-level within a dataset and conclude that a few well-chosen classes used as pre-training could build a network that is well-initialized that even with 100 times less samples, we could learn the other classes. We also provided some practical guidelines for a CNN engineer to adopt. After performing similar experiments on popular imaging datasets and medical datasets, we made similar serendipitous observations.

Acknowledgments: This work was supported in part by ARO grant W911NF1410371. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of ARO.

6. REFERENCES

[1] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842*, 2014. 1

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105. 1

[3] Bao-Qing Li and Baoxin Li, "Building pattern classifiers using convolutional neural networks," in *Neural Networks, 1999. IJCNN'99. International Joint Conference on*. IEEE, 1999, vol. 5, pp. 3081–3085. 1

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587. 1

[5] Victor Escorcia, Juan Carlos Niebles, and Bernard Ghanem, "On the relationship between visual attributes and convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1256–1264. 1

[6] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328. 2

[7] 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, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, pp. 1–42, April 2015. 2

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015. 2

[9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 2

[10] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 473–480. 2

[11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*. Granada, Spain, 2011, vol. 2011, p. 5. 2

[12] T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*, February 2009. 2

[13] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009. 2

[14] Li Fei-Fei, Rob Fergus, and Pietro Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007. 2

[15] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012. 2

[16] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015. 3