NEURAL DATASET GENERALITY

RAGAV VENKATESAN

VIJETHA GATUPALLI

BAOXIN LI

SIFT

HOG

ALL ABOUT THE FEATURES

GABOR

DAISY

AlexNet

GoogleNet

CONVOLUTIONAL NEURAL NETWORKS

VGG-19

ResNet

FEATURES COMES FROM DATA

- PCA
- Dictionaries
- Neural Networks

We know/posit that these features are the best representations of data for the dataset that we are currently concerned about.

What about off-the-shelf neural features?

OFF-THE-SHELF FEATURES

Extract features from a large dataset such as ImageNet

Make the feed forward network public.

Download off-the-shelf networks.

Extract features on user dataset.

Train a new classifer on top.

OFF-THE-SHELF FEATURES

Extract features from a large dataset such as ImageNet

Make the feed forward network public.

Download off-the-shelf networks.

Extract features on user dataset.

Train a new classifier on top.

Is there a guarantee that the downloaded features can capture the intricacies and idiosyncrasies of the user dataset?

OFF-THE-SHELF FEATURES

Most often not!

But we roll with it. – Because it works.

Is there a guarantee that the downloaded features can capture the intricacies and idiosyncrasies of the user dataset?

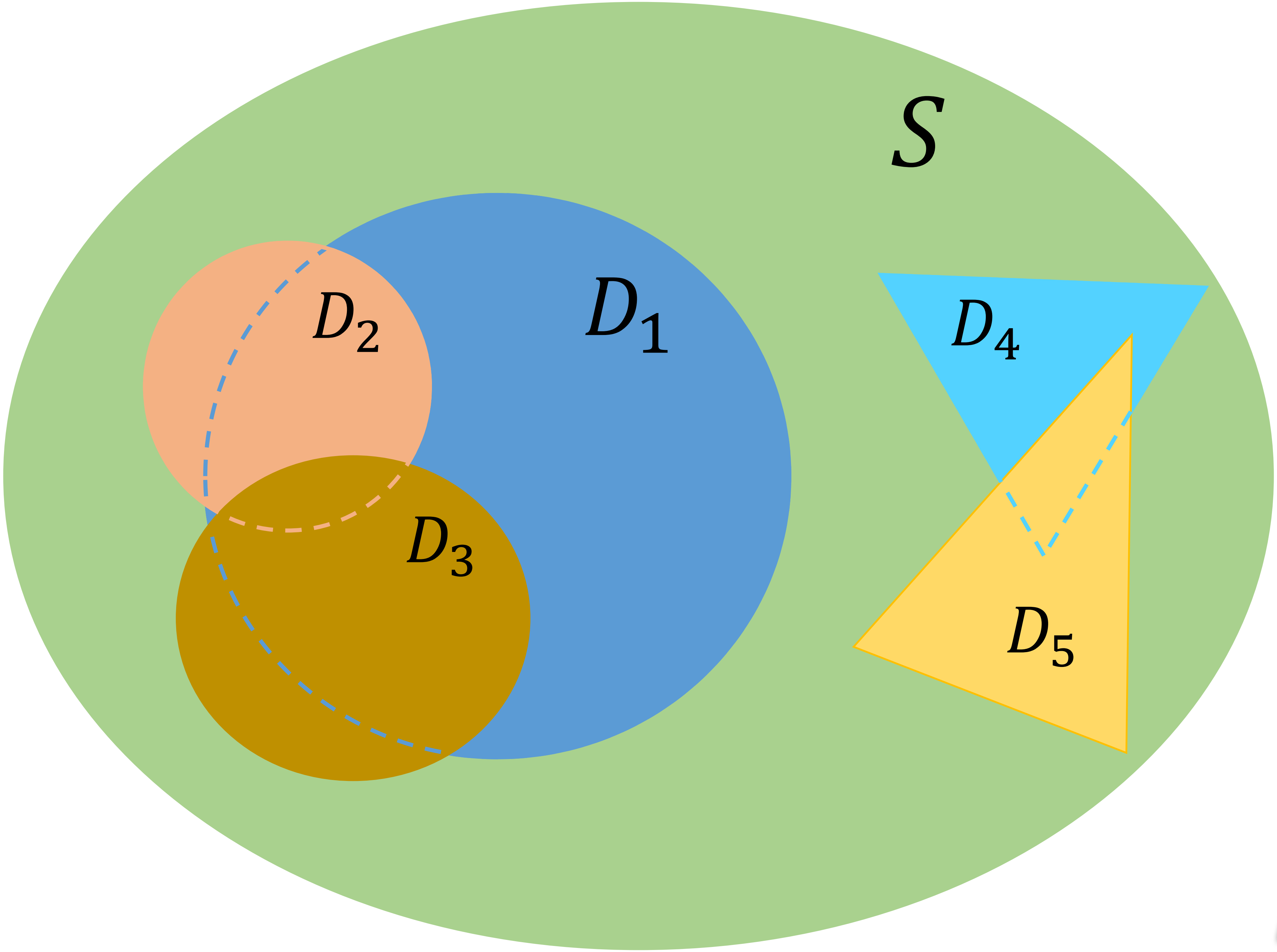
The ubiquity of downloaded CNNs
Unquestioned performance of networks trained on ImageNet

One network fits all.

But does it?

ATOMIC STRUCTURES

- CNN filters take some shapes due to the entropy of the dataset.
- Some datasets have some unique idiosyncrasies that show up as atomic structures.
- These may be edges and Gabor filters in the first layers and so on.


S
D_1
D_2
D_3
D_4
D_5

A GENERALITY RANKING METRIC

- Generality is not a rankable concept.
  - Due to the overlapping nature of feature expressions, representations aren't usually nestable or complete.
  - Generality is only a relative concept.

- Can we use the neural training procedure and dataset performance to measure dataset generality ? - Yes.
  - Very close corollary to network transferability and remembrance. [1].

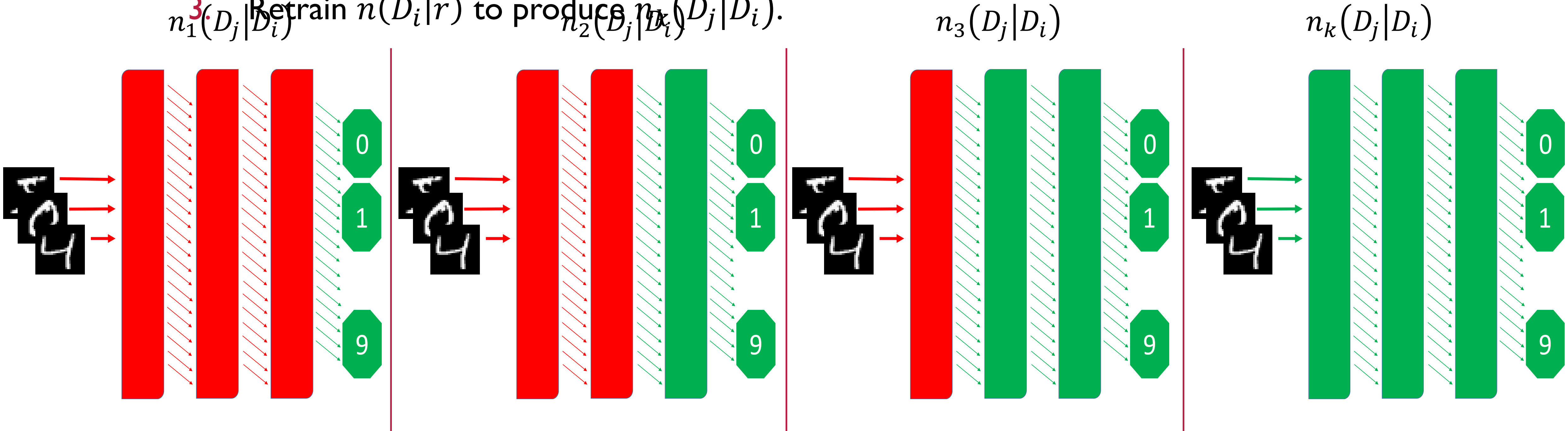
[1] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," in Advances in Neural Information Processing Systems, 2014, pp. 3320–3328.

DETOUR …. OBSTINATE LEARNING.

- An obstinate layer is a layer whose weights are not allowed to update during training.
- Gradients are simply ignored.
- An obstinate layer and all the layers that feeds into the obstinate layers must all be frozen.
  - Downloading a network and training only the softmax layer.
  - Layer-wise pre-training.
  - Dropouts (not exactly but similar).

EXPERIMENT SETUP

1. Consider two datasets D_1 and D_2.
2. Initialize a network with random weights and train with D_i.
   - This network is called the *base network* and is represented by $n(D_i|r)$.
3. Retrain $n(D_i|r)$ to produce $n_k(D_j|D_i)$.



GENERALITY METRIC

- Performance of $n(D_j|r)$ is $\Psi(D_j|r)$.
- 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 layer k is:

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

Performance that is achieved by D_j using,

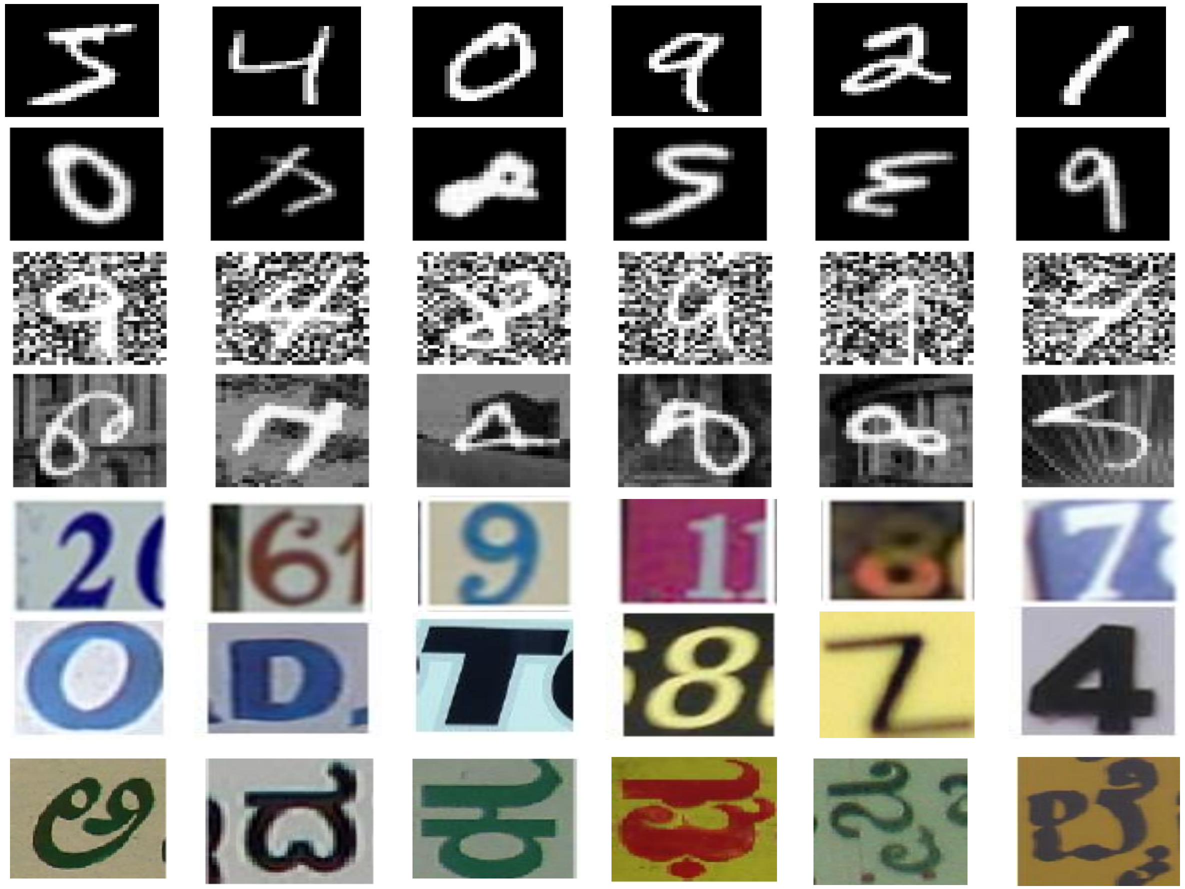
- $N - k$ layers worth of prejudice from D_i
  - k layers worth of features from D_i
- k layers of novel knowledge from D_j

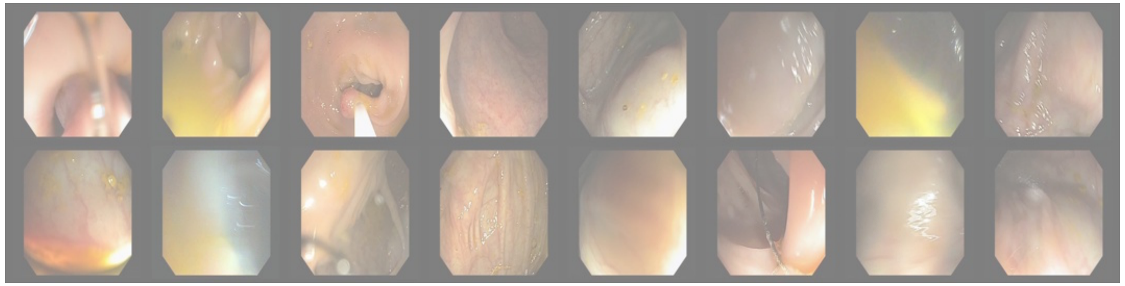
$$g_k(D_i, D_j) = \frac{\Psi_k(D_j | D_i)}{\Psi(D_j | r)}$$

PROPERTIES OF THIS GENERALITY METRIC

- $g_k(D_i, D_j) > g_k(D_i, D_l) \rightarrow$ 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 .
- $g_k(D_i, D_i) \geq 1 \; \forall k$.
- $g_k(D_i, D_j)$ for $i \neq j$ might or might not be greater than 1.

DATASETS CONSIDERED

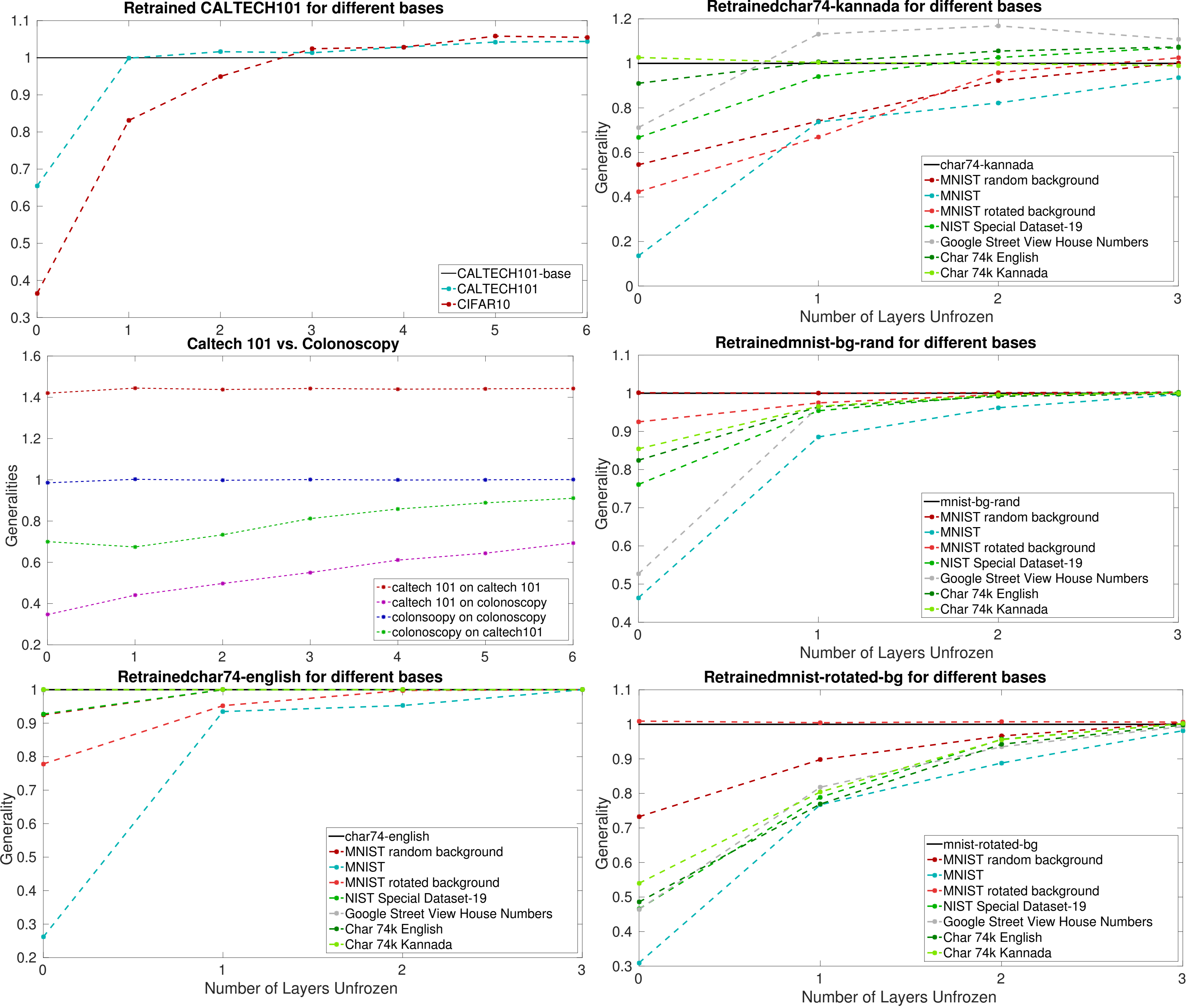


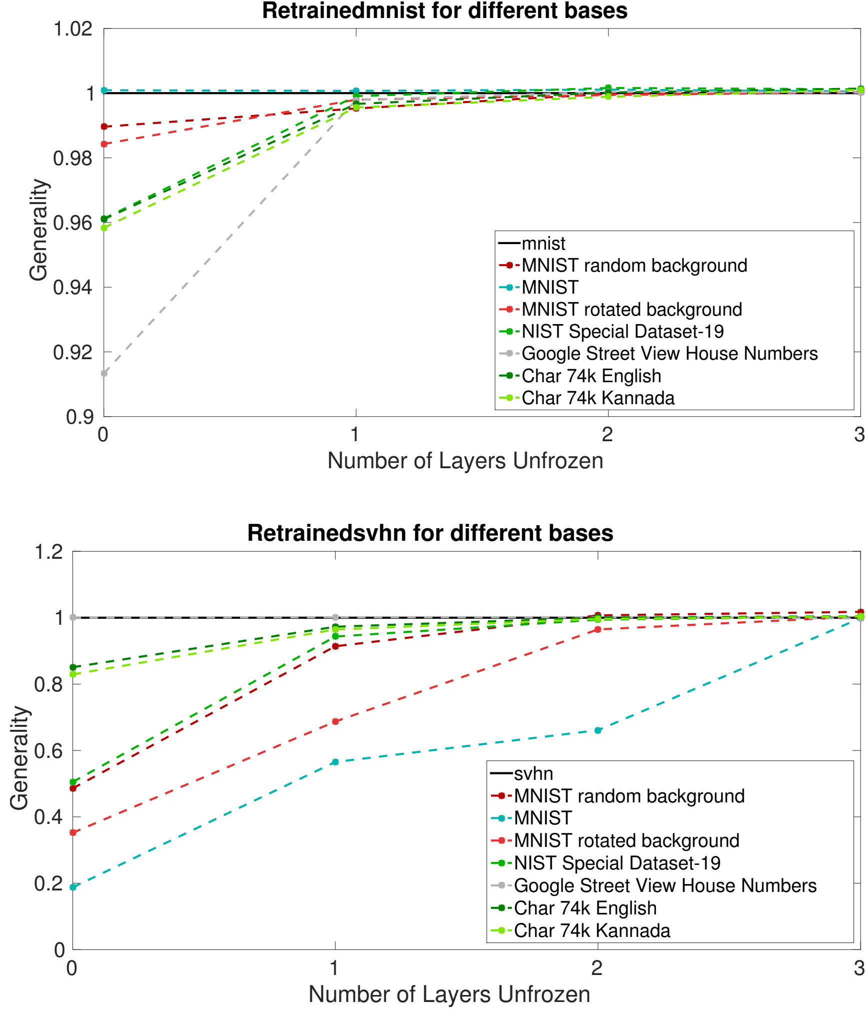


REFERENCES TO DATASETS

1. 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. - MNIST and all its rotations.
2. 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.
3. 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.
4. Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009.
5. 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.

SOME INTERESTING RESULTS





Generality curves for each dataset as base against all the other comparing datasets.

PARSING THE GRAPHS

SOME SURPRISING RESULTS

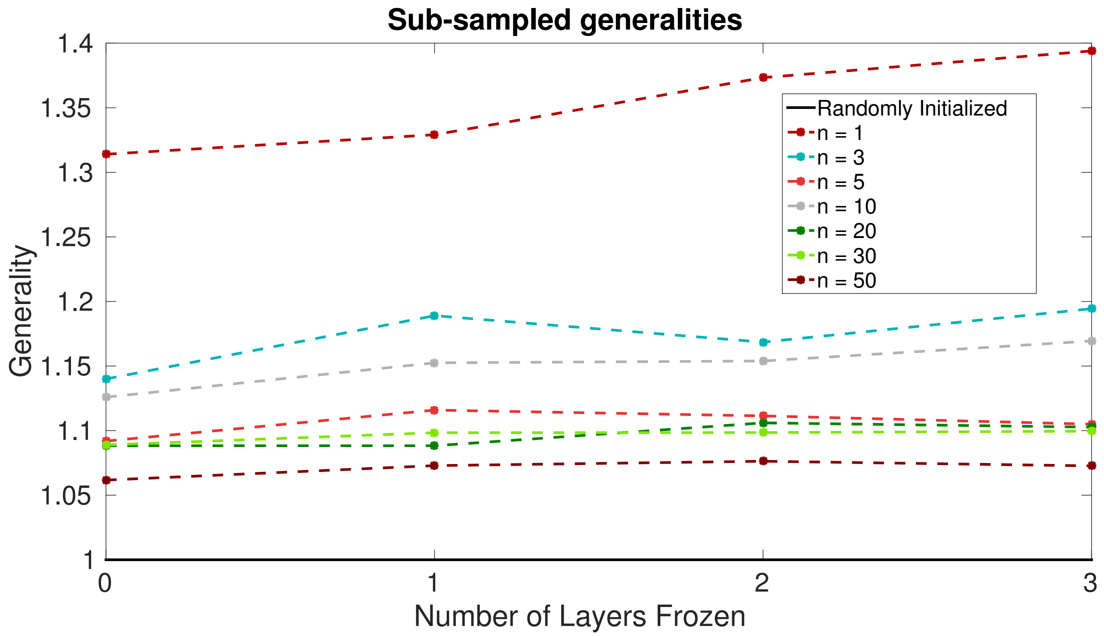
- No dataset is qualitatively the most general.
- MNIST dataset is the most specific.
  - Rather, MNIST dataset is one that is generalized by all datasets very highly at all layers.
  - MNIST dataset actually gives better accuracy when prejudiced with other datasets than with random intits or even when prejudiced with itself !!
  - This is a strong indicator that all datasets contain all atomic structures of MNIST.
- English and Digits are more general than Kannada !!
  - While MNIST and MNISt-rotated are not general, other MNIST with backgrounds, Google SVHN, NIST and Char74-English are all more general than Char74-Kannada.

INTER-CLASS DATASET GENERALITY

- D_i and D_j need not be entire datasets but can also be just disjoint class instances of the same dataset.
- For instance, we divided the MNIST dataset into two parts.
  - MNIST $[4, 5, 8]$ (base) and MNIST $[0, 1, 2, 3, 6, 7, 9]$ (retrain).
- 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.

INTRA-CLASS GENERALITY - RESULTS

- 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.



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

INTRA-CLASS GENERALITY - RESULTS

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

- 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 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.

MORE RESULTS…

- Once initialized with a general enough subset of classes from within the same dataset, the generalities didn't vary among the layers.
- The more the data we used, more stable the generalities remained.
- 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.

Neural Dataset Generality

Ragav Venkatesan, Vijetha Gattupalli, Baoxin Li

(Submitted on 14 May 2016)

Often the filters learned by Convolutional Neural Networks (CNNs) from different datasets appear similar. This is prominent in the first few layers. This similarity of filters is being exploited for the purposes of transfer learning and some studies have been made to analyse such transferability of features. 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.
It is curious that 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 among datasets from their accuracies on transferred filters. We applied this metric on several popular character recognition, natural image and a medical image dataset, and arrived at some interesting conclusions. On further experimentation we also discovered that particular classes in a dataset themselves are more general than others.

Comments: Long version of the paper accepted at IEEE International Conference on Image Processing 2016
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**
Cite as: arXiv:1605.04369 [cs.CV]
(or arXiv:1605.04369v1 [cs.CV] for this version)

Download:

- PDF
- Other formats

(license)

Current browse context:
cs.CV
< prev | next >
new | recent | 1605

Change to browse by:
cs

References & Citations
- NASA ADS

DBLP - CS Bibliography
listing | bibtex
Ragav Venkatesan
Vijetha Gattupalli
Baoxin Li

Bookmark (what is this?)

ragavvenkatesan / generality

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Code for the paper, Neural Dataset Generality by Ragav Venkatesan, Vijetha Gatupalli and Baoxin Li — Edit

9 commits | 1 branch | 0 releases | 1 contributor | MIT

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

ragavvenkatesan updated to match samosa updates. Latest commit 8665706 on Feb 29

.gitignore	Added Gitignore	11 months ago
License.md	Initialize code for GitHub Push.	8 months ago
README.md	Update README.md	8 months ago
__init__.py	updated to match samosa updates.	7 months ago
dataset_setup.py	Initialize code for GitHub Push.	8 months ago

README.md

To run the code first download the Samosa Toolbox (https://github.com/ragavvenkatesan/Convolutional-Neural-

This work was supported in part by ARO grant W911NF1410371.

Fin.