Visualizing and Understanding Convolutional Networks with Logpolar Images

Prahal Arora
Department of Computer Science
UC San Diego
A5321950
prarora@eng.ucsd.edu

Abstract

In this paper, we address two tasks. First, we train a VGG-16 network from scratch on log-transformed images in ImageNet dataset (ILSVRC 2012), without any pre-training or fine-tuning on ImageNet weights. We evaluate it's performance on image classification task. Second, we visualize the highest activations in the network that gives insight into the function of CNN layers for classification of log-polar transformed images.

1 Introduction

Mapping between retina and the cortex in human visual system is a log-polar transformation that creates cortical magnification of central representations. In a recent study to understand the central and peripheral vision for scene recognition by Panqu et al.[2], a neurocomputational model was developed to model the results of behavioral experiments of Larson and Loschky (2009), which aimed to study the impact of scene classification accuracy with varying fraction of viewable area. Author found that using logpolar transformed images to account for cortical magnification makes the modeling results closer to the behavioral data. To get a deeper insight into the effectiveness of log-polar transformation on the images, we train the model from scratch using the log-polar transform, where we just used VGG net as our backbone model without doing any pre-training like previous works[2]. We do this to study the impact of cortical features when expanded in pixel real-estate, and to see if networks learns to extract useful information from it.

In addition to that, we try to visualize the features learned in this way to observe whether there are differences in the features. This is essentially helpful when we want to gain insight into the internal operation and behavior of these complex models trained on log-polar images, or how they achieve their performance. In this work we use a visualization technique first proposed by Zeiler et al.[1] that reveals the input stimuli that excite individual feature maps at any layer in the model. It also allows us to observe the evolution of features during training. The visualization technique we propose uses a multi-layered Deconvolutional Network, to project the feature activations back to the input pixel space. However, we found that this visualization technique doesn't result in crisp and sharp visualization for VGG-16 network. We use a slightly modified visualization technique proposed in [3] to get a clean and crisper mapping of activations..

2 Experiments

2.1 Training the VGG16 on log transformed imagenet

We used PyTorch for developing code for this experiment. We trained a vanilla VGG-16 from scratch on log-polar version imagenet without Batch Normalization for 100 epochs, using the prescribed

dataset	top1 accuracy	top5 accuracy
normal imagenet	71.56	90.38
log-polar imagenet	65.724	86.613

Table 1: Top1 and Top5 accuracy results of two datasets trained on VGG-16

learning rate for imagenet dataset. Instead of converting the entire imagenet dataset into log-polar equivalent, we applied the log-polar transformation on the fly when reading the data from the disk. Counter-intuitively, this turned out to be a very efficient operation as PyTorch allows various processes to read data in parallel, also allowing prefetching of the data when the training of a batch is in progress. Hence, the computational overhead of log-polar transformation operation is negligible. We used openCV implementation for log-polar transformation. This was trained on 2 GPUs in a data parallel manner. Training of 1.3M images took 7 days to converge. The code for this project is shared in github repository `https://github.com/prarora/logpolar_imagenet`.

2.2 Visualizing the dominant activations

Code for this project was developed in PyTorch, using 2 GPUs in a data parallel fashion. For higher layers of the network the method of Zeiler and Fergus[1] fails to produce sharp, recognizable, image structure.The visualization technique[3] that we used was slightly different than the one proposed in [1]. In [3], authors propose a modification of the 'deconvnet', which makes reconstructions significantly more accurate, especially when reconstructing from higher layers of the network. The 'deconvolution' is equivalent to a backward pass through the network, except that when propagating through a nonlinearity, like ReLu, we set all the activations to zero, that were less than zero during the forward propagation through the ReLu layers. This proposed method doesn't use Unpool switches, and they call this method guided backpropagation. Interestingly, unlike the 'deconvnet', guided backpropagation works remarkably well without Maxpool switches for unpooling, and hence allows us to visualize intermediate layers as well as the last layers of our network . In a sense, the bottom-up guided signal in form of the pattern of 'ReLU activations of the forward propagation'substitutes the Maxpool switches.For a given layer number and the channel number in VGG-16, we monitor the highest neuron activation in the feature map by forward propagation of 50000 images in the imagenet valdiation set. We keep track of 9 images that resulted in highest activation. For each of these 9 images, we set all other activations in the chosen feature map, except of the highest activation to 0. We do this experimentation twice, once with the weights trained on normal imagenet dataset, and one with weights trained on log-polar imagenet from the scratch. The code for this project is shared in github repository `https://github.com/prarora/Decov-VGG16-PyTorch`.

3 Result and Analysis

3.1 Accuracy

Table 1 encapsulate the top1 and top 5 accuracy of VGG-16 trained on normal imagenet dataset and log-polar transformed imagenet dataset from scratch. The results reported are achieved on imagenet validation set for ILSVRC 2012 which contains 50000 images. As you can observe, the results achieved on log-polar imagenet are competitive to normal imagenet training. This suggests that log-polar 'warping' of images contains useful features that could be extracted by the hierarchy of CNN layers, and later used by fully connected part for the image classification. Also, another point worth noting is that the hyper-parameters used for log-polar training were tuned on normal imagenet training. This means that there might be some potential for higher accuracy on log-polar imagenet with hyper parameter tuning.

3.2 Visualization

Figure 1 shows the top 9 activation for 10 random channels(feature maps) in layer 12 of VGG-16. The receptive field of conv12 is 164 x 164. The activations are mapped back to image space. These activations are corresponding to the normal imagenet dataset. We also on right the actual images that caused the maximal activation and the category name.

One can observe that their is strong grouping in the top 9 activation, for eg. the top 9 activation in row 1, column 1 shows strong inclination to bugs, flies and spiders. Similarly top 9 activations in row 1 and column 2 show strong response to checks and squared/round pattern. Also, it's worth noticing that these are high level features, such as head of a green reptile, dogs face, birds face and alike.

Figure 2 and 3 shows the top 9 activation for 10 random channels(feature maps) in layer 12 of VGG-16 trained on log-polar imagenet. If we directly look at the log-polar features, it's difficult to find an obvious consistency in top9 activations. However, if we apply the inverse log-polar features and images, the consistency becomes obvious. Therefore, we apply the inverse log-polar transform to these activations and images for the purpose of visualization. Beneath every such image, we also show the log-polar features and the log-polar image that cause the corresponding maximal activation, because this was the image that the network actual saw during training.

One can observe that their is strong grouping in the top 9 activation, for eg. the top 9 activation in row 1, column 1 shows strong inclination to human faces. Similarly top 9 activations in row 1 and column 2 show strong response to big dials and circular patterns in the image. However, the grouping is not as strong as normal imagenet classification Also, it's worth noticing that due to the nature of log-polar transformation, a lot of information is encoded inside a small region away from the image center. Therefore, for the same receptive field, the inverse log-polar transformed features show responses spanning much more pixels. Another observation is that cortical magnification area has no strong feature response, suggesting that this part of the log-polar images are not very useful.

4 Conclusion

We have shown the effectiveness of training a deep net on log-polar images. The top1 and top5 accuracy achieved on log-polar imagenet demonstrates the ability of this transformation to possess discriminative features needed for image classification task.
Going further, we visualized the features of a deep net trained on log-polar imagenet from scratch to understand the feature extraction process for this special transformation and compare it with the corresponding maximal activations for normal images.

References

[1] Matthew D. Zeiler & Rob Fergus (2013) Visualizing and Understanding Convolutional Networks, *CoRR*, bs/1311.2901

[2] Wang, Panqu and Cottrell, Garrison W. (2017) Central and peripheral vision for scene recognition: A neurocomputational modeling exploration *Journal of Vision* 0.1167/17.4.9

[3] Jost Tobias Springenberg and Alexey Dosovitskiy and Thomas Brox and Martin A. Riedmiller (2014) Striving for Simplicity: The All Convolutional Net*CoRR* abs/1412.6806.

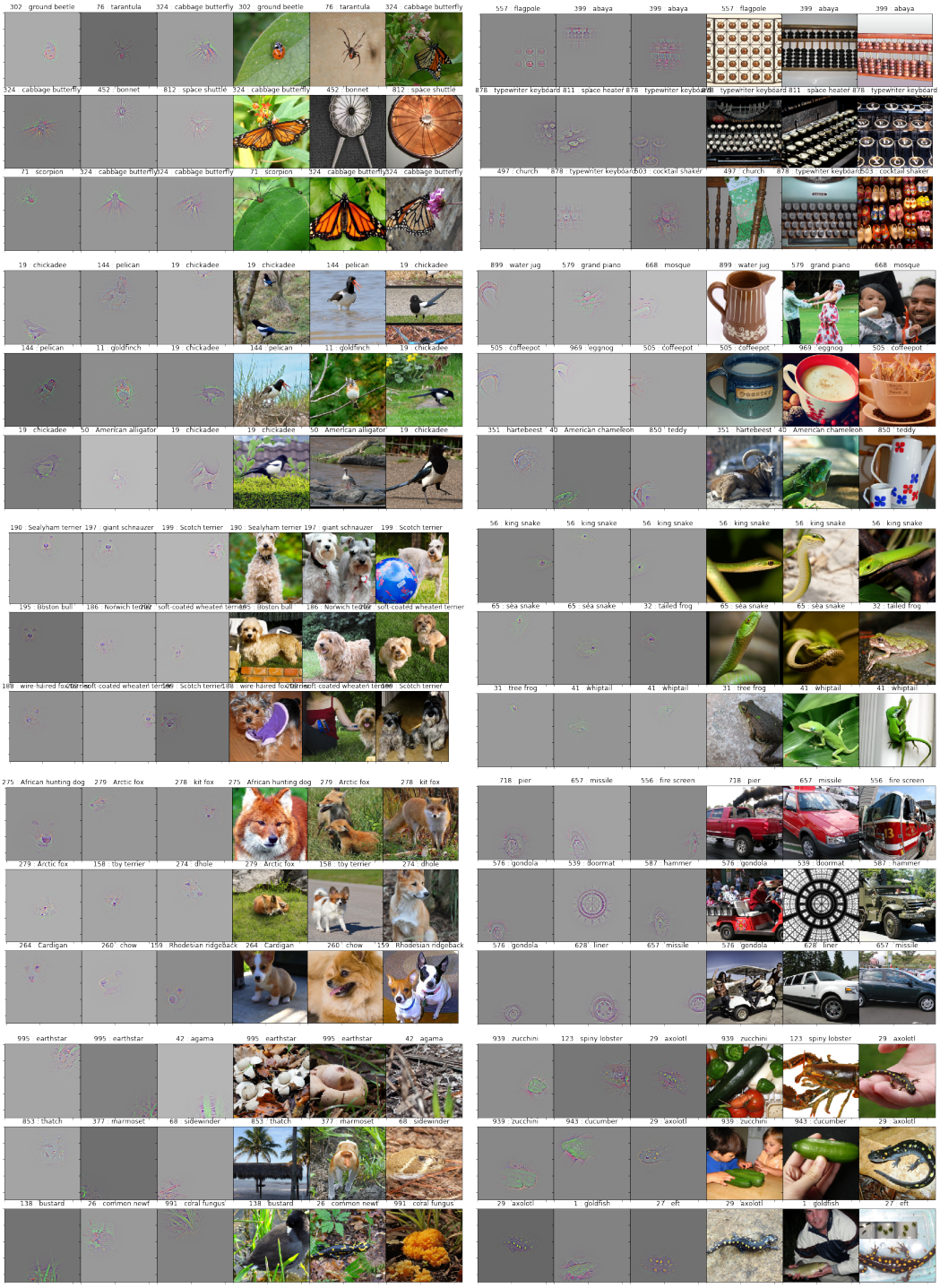

Figure 1: Visualization of top 9 activation for random feature maps for layer 12 of VGG16 and corresponding images from validation set

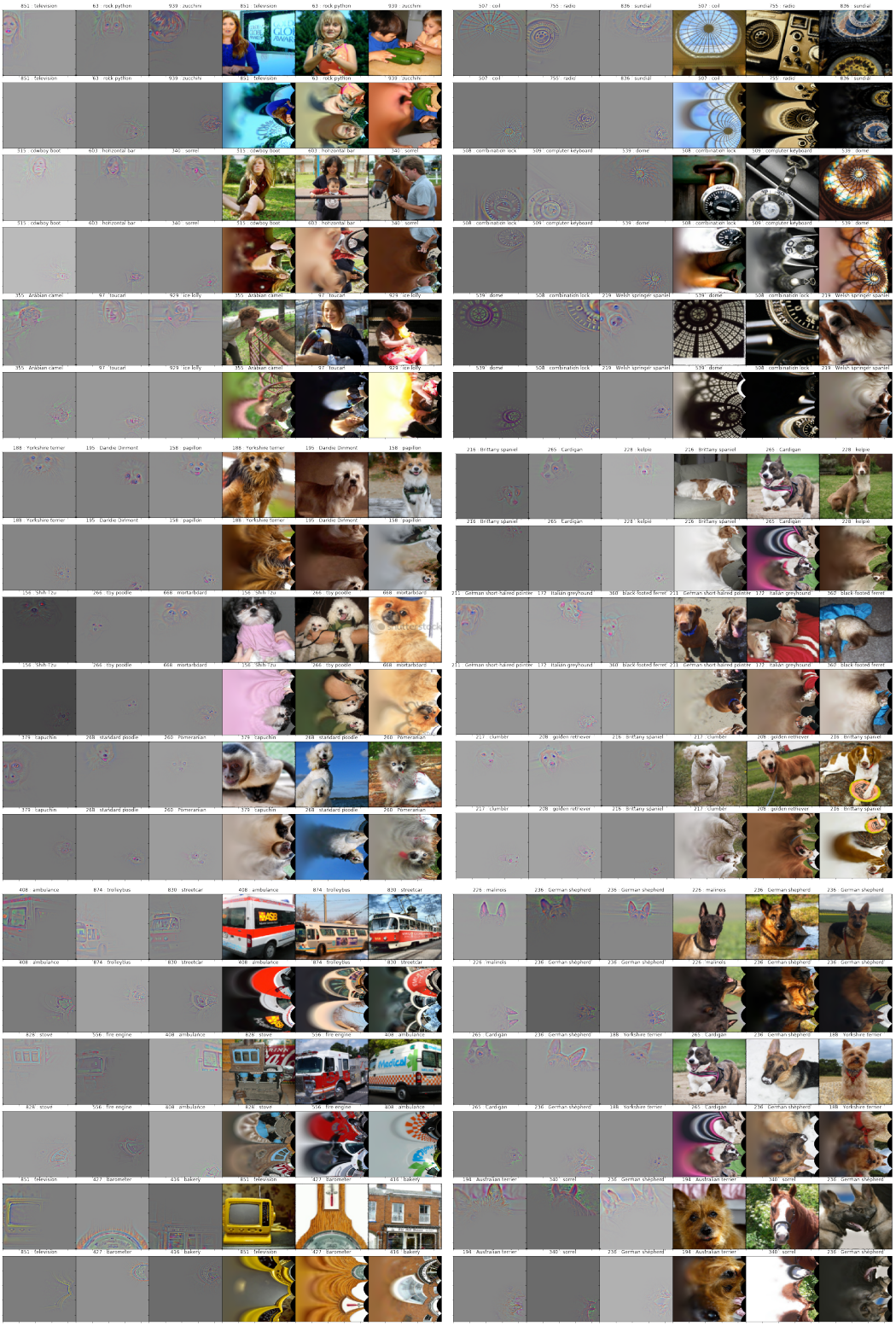

Figure 2: Visualization of top 9 activation for random feature maps for layer 12 of VGG16 and corresponding images from validation set of log-polar imagenet

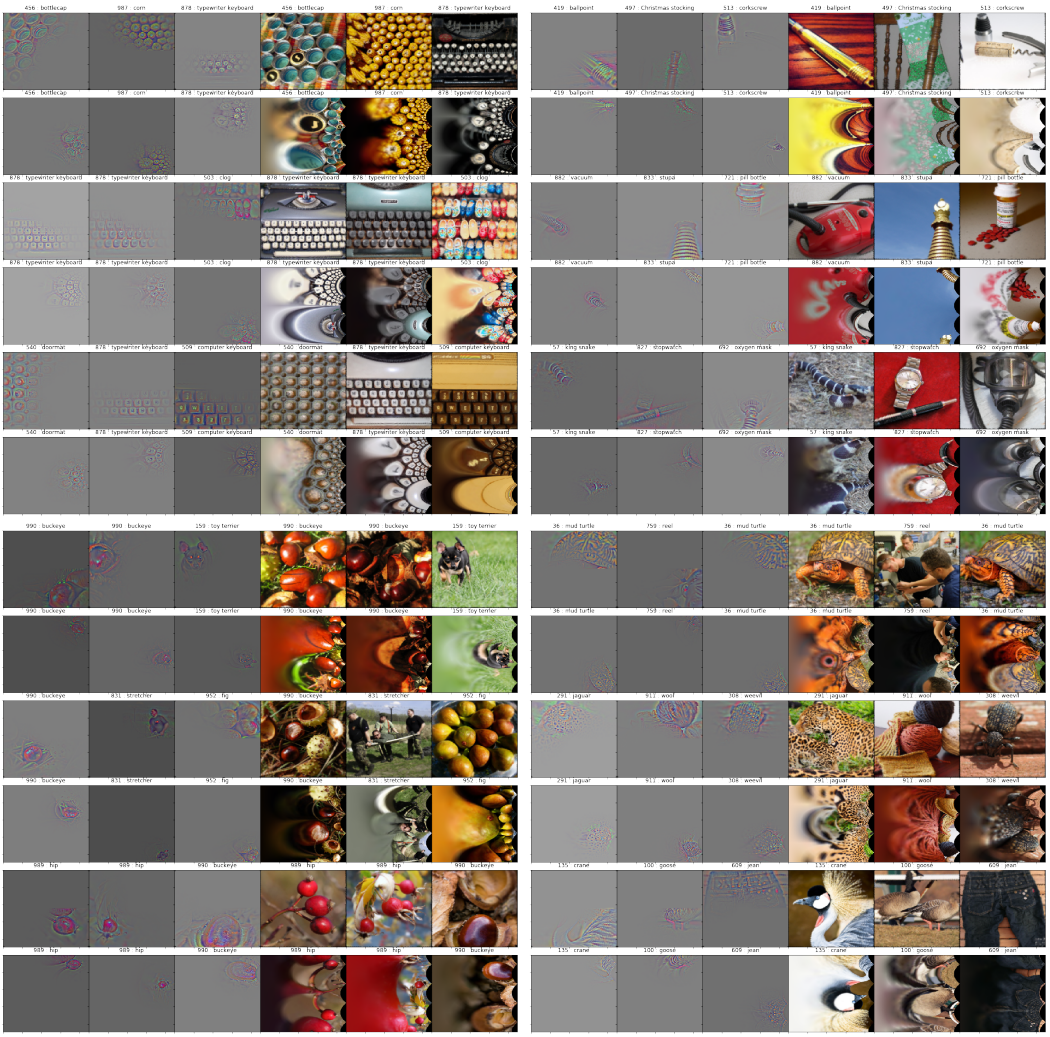

Figure 3: Visualization of top 9 activation for random feature maps for layer 12 of VGG16 and corresponding images from validation set of log-polar imagenet