CNN Image Recognition Architecture Simplification using Patch-Based Data Reduction Techniques

Jiying Zou	Rui Yan	Yuan Liu
Statistics	CME	Data Science
Stanford University	Stanford University	Stanford University
jiyingz@stanford.edu	ruiyan@stanford.edu	linda921@stanford.edu

1 Introduction

Convolutional Neural Networks (CNN) have risen in popularity as the forefront technique of image recognition technology. CNNs are so successful because they adapt neural networks' non-linear interpolation abilities to image datasets, providing versatility to learn a hierarchy of filters useful for image classification tasks. However, as the race to set new benchmark heights continues, one significant issue is often overlooked. Neural networks are not run for free. By design, they require massive datasets in order to effectively train and yield generalizable results, and for CNNs even more so. Likewise, processing through an image requires runtime storage of an amount of weights that increases with image size. Massive amounts of energy and physical storage is needed to run a state-of-the-art CNN. While there exist many research efforts in the direction of reducing computational intensity for neural networks, most of the time a lack of emphasis on this issue is perhaps due to "a lack of familiarity with the approaches to estimate energy consumption" (Martin et al, 2018).

There are multiple approaches to this conundrum. One method would be additional educational requirements for engineers and researchers in the field. A second and more practical approach is to capitalize on current knowledge and alter neural network architectures in ways to reduce time and energy consumption from running models. We aim to generate insights in the latter through our efforts in this project.

2 Literature Review

Deep neural models based on Convolutional Neural Networks (CNNs) have led to stunning performance on complex computer vision tasks including image classification and object detection. Nevertheless, their decision-making process remains a mystery due to their lack of decomposability into individually intuitive components (Z.C Lipton, 2016).

There has been a number of recent works that aim to add interpretable elements to state-of-art CNNs. Pinheiro & Collobert (2014) proposed a technique that adds explicit labeling of pixels before integrating to a image-level decision. Since pixel labels are derived from the whole image, the process of inferring which pixels are important for classification remains a difficult question. Zhou et al. (2015) discovered that using class activation mapping together with the global average pooling layer allows a classification-trained CNN to both classify the image and localize class-specific image regions. Though such approaches highlight important regions for discrimination, it fails to show how exactly each image patch plays a part in final decision-making. A novel step in this direction proposed by Brendel and Bethge (2019) utilizes a new DNN architecture inspired by previous bag-of-feature (BoF) models to not only yield performance comparable to state-of-art DNNs, but also shed light on how small image patches' evidences are integrated to reach an image-level decision.

Traditionally, DNNs work by aggregating local features via learned convolutions followed by spatial pooling. Successive application of these "convolutional layers" results in a "hierarchy of features" that integrate low-level information across a wide spatial extent to form high-level information. Current deep CNNs perform global integration of information, that is, their performance requires "seeing the forest for the trees" (Hsu, Zhang & Yang, 2016). However, it was suggested that by using BagNets proposed by Brendel and Bethge (2019), a non-linear model is no longer needed to integrate local features into a global representation; instead, we can now classify an image solely based on the occurrences of small local image features without taking into account their spatial ordering. Through limiting the receptive field size of the convolution layers, each element of the convolved tensor now holds the local feature of an individual image patch. After feeding convolved tensor into an average pooling layer then the fully-connected layer, class evidence used for the classification is obtained.

3 Datasets

A flowers dataset from Kaggle is used, containing 4242 labeled flower images across 5 categories: daisy, dandelion, rose, sunflower, and tulip. The images were originally sourced from Flickr, Google images, and Yandex images. Data is divided into a 70%-20%-10% train-validation-test set split, yielding 2968 train images, 848 validation images, and 426 test images. The number of train/validation/test images in each category is shown in Fig. 1.

	daisy	dandelion	rose	sunflower	tulip
train	538.0	693.0	548.0	513.0	676.0
val	154.0	198.0	156.0	147.0	193.0
test	77.0	99.0	79.0	74.0	97.0

Fig.1 The train-validation-test split of flowers data with five output categories.

All images were resized to 224x224 pixels to fit ResNet-50 and BagNet-33 expectations. Data augmentation including random crop and horizontal flip of training data was employed to improve model performance and generalizability. Images were then converted to torch tensors and normalized to standard scale, which helps accelerate the training process and avoid local optima.

4 Methods

Baseline models are all pre-trained on ImageNet. Models have modified final layers to match our number of output classes.

4.1 ResNet

The first baseline model is a ResNet-50 (He et al., 2016). ResNet is a well-established state-of-the-art CNN architecture, and ResNet-50 is a specific 50-layer variant. We chose this as a baseline because we want to compare results to benchmark performance and verify BagNet's competence. Also, BagNet uses ResNet-50 models, so the choice maintains model comparability. ResNet layers are organized into 4 main structures, each containing several blocks with 3x3 and 1x1 convolutional layers, with a pooling layer at the end. These convolutional layers infer filters by moving patch-by-patch through the entire image, learning basic structures (such as edges and corners) on the lower levels and more complex features (such as eyes or leaves) at the deeper stages. Inferred features are then used to predict class. ResNet uniquely uses "identity connections" to fix the vanishing gradient problem, enabling superior performance.

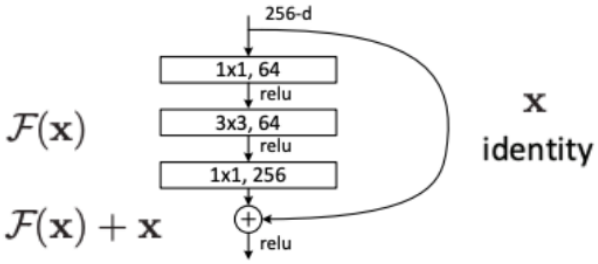

Fig. 2. A single convolutional block of the ResNet architecture, displaying the "identity connection" used to fix the vanishing gradient problem. Image credit: https://arxiv.org/pdf/1512.03385.pdf

4.2 BagNet

We use BagNet-33, a BagNet variant with 33x33 patch sizes, as another baseline model and run experiments upon this architecture. BagNets are inspired by bag-of-feature (BoF) models, which generate identifying sparse feature distributions per class and compares observation-specific feature distributions to them to make predictions. For image recognition with BagNets, this translates into running a modified ResNet-50 over each patch of size *qxq* to obtain patch-specific *kx1* class evidence vectors (for *k* classes). The ResNet is modified by replacing all 3x3 convolutional layers with 1x1 convolutions, effectively limiting the top-level receptive field to only the patch itself. Patch class evidences are linearly combined (simple average aggregation) and logit transformed to predict probabilities that an image is of each class. Note that using this heuristic, the classification is patch-order-agnostic, i.e. patch spatial locations and relationships do not play a factor in making final predictions.

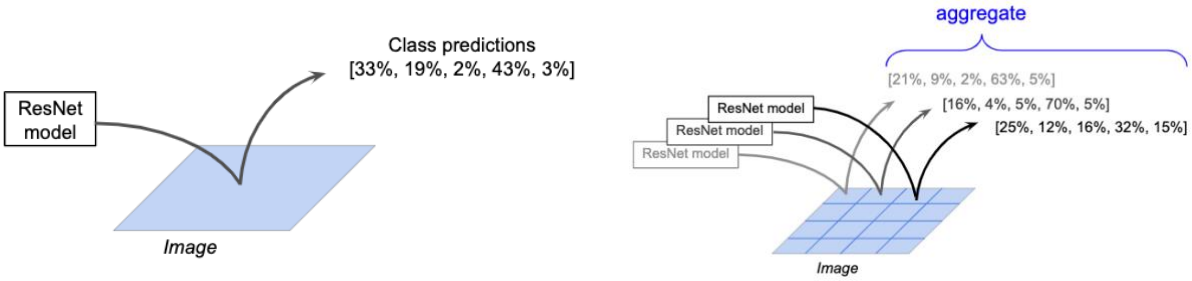

Fig. 3. Left: ResNet model heuristic. Right: BagNet model heuristic. ResNet makes one set of class predictions, while BagNet makes a final class prediction by averaging patch class evidences.

4.3 Evaluation Metrics

Classification accuracy and average cross-entropy (were used as model evaluation metrics.

$$\text{Average Cross-Entropy} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_k^{(i)} \log \hat{y}_k^{(i)} \text{ for } k \text{ classes}$$

5 Experiments

We want to explore whether all observation patches are needed in order to make a decent prediction. Thus, the main idea of the experiments is to "blackout" certain patches during the testing phase by setting their weights to 0 so that they are not considered in the final aggregation.

Four experiments were considered. In the first, we "blackout" every other patch and make predictions. In the other three, we randomly "blackout" 75%, 50%, and 25% of all patches, respectively. The visualizations below provide a sample of what effect "blackout" is simulated to have on the images.

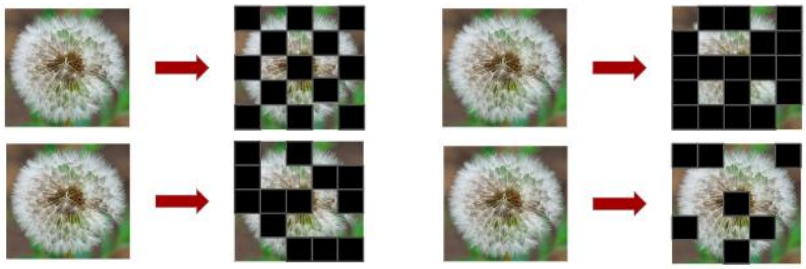

Fig. 4. Far left: Experiment 1 (alternating "blackout"). Middle left: Experiment 2 (random 75% blackout). Middle right: Experiment 3 (random 50% blackout). Far right: Experiment 4 (random 25% blackout).

6 Results and Discussion

Comparing BagNet-33 with ResNet-50, we see that BagNet-33 shows comparable performance in both prediction accuracy and loss to ResNet-50 (Table 1 and Fig. 5), suggesting that CNN might care about patch details/patterns in an order-agnostic way. Moreover, Random 50% blackout is found to be comparable to alternating blackout, further backing up the order-agnostic theory. It also seems possible to randomly dispose of ~25% of patch results on an image and still maintain prediction accuracy comparable to that of full BagNet-33. This finding implies that DNNs are using weak and local statistical regularities to make decisions. For considerations of memory and computational resources, this may be a good news in that we can now use less resources to achieve similar performance. However, this also means that the state-of-art DNNs are fundamentally different from the true neural networks in human visual system. When blacking out 25%, 50% of patches, humans are still capable of inferring the correct class label. The reason behind this resilience to blacking out lays in the fact that humans utilize holistic features and global shape integration to recognize objects. Such method allows us to interpolate and fill in the missing puzzle pieces even when a large portion of information is missing. For the purpose of both pushing performance of DNNs to a new extreme and generating DNNs qualitatively comparable to human visual system, we should encourage the current network architectures - either through different architecture design or training procedure - to learn more holistic features that can fully utilize the inherent causal relationships between different parts of the images. In addition, we observed that dropping out patches causes loss to increase (and accuracy to decrease) in a non-linear fashion.

	ResNet-50	BagNet-33	BagNet-Alternating "blackout"	BagNet-Random 25% "blackout"	BagNet-Random 25% "blackout"	BagNet-Random 75% "blackout"
Train Acc/Loss	0.9211/0.2517	0.9288/0.2799				
Test Acc/Loss	0.9093/0.2847	0.8821/0.3745	0.8293/0.5761	0.8827/0.4123	0.8327/0.5445	0.8327/0.5445

Table 1. Train and test accuracy/loss

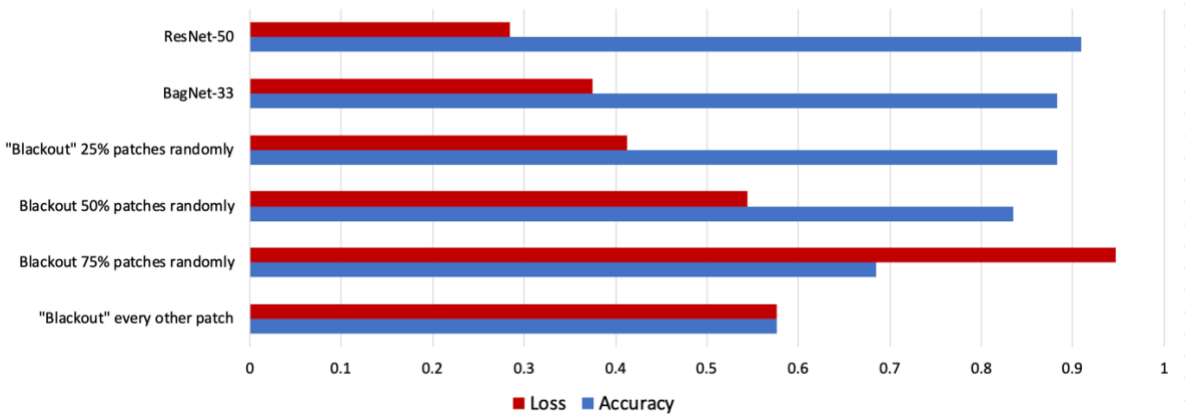

Fig. 5. Test accuracy and loss for each model

7 Conclusion

This project provides proof-of-concept that not only is it possible to run CNNs more time- and space-efficiently by utilizing the BagNet architecture, it is also not necessary to retain all of the patches of an image when making a class prediction. This further cuts down on computational resource necessities. In fact, around 25% of patches in an image can be safely ignored while maintaining comparable prediction accuracy and loss. Interestingly, dropping out 50% of patches by alternating "blackout" performed comparably to random "blackout", providing evidence that class inference using patches are patch-order-agnostic. On a high level, we have reproduced BagNet-33's comparable performance to ResNet-50, produced supporting evidence for insight into what CNNs care about, and created proof-of-concept for a new way to save computational resources when running CNNs for image classification.

8 Future Work

In the current work, we experimented with different aggregation method of BagNet-33 on the flowers dataset. To improve the generalizability of our results, several approaches could be used. First, it is worthy of investigation to apply the same methods to other BagNet models, namely BagNet-17 and BagNet-9, to see if the conclusion still holds. Second, attempts need to be made to identify current dataset bias. One way to achieve this is to utilize the Gradient-weighted Class Activation Mapping (Grad-CAM) proposed by Selvaraju et al (2016) to localize important regions used in prediction. Third, the methods should be applied to larger datasets (e.g. ImageNet) to see if the results can be replicated. Last, stochasticity and robustness analysis should be performed such that the performance of our modified BagNet-33 does not deteriorate on new independent but similar dataset. For future directions, we could test out other aggregation schemes, such as nonlinear aggregations or weighting patch results by proximity to image center. As for industrial applications, the current results are at the stage of a proof-of-concept, and how to utilize "blackout" to save memory and computational resources demands further investigation.

9 Contributions

Jiying contributed to code and data setup, ResNet-50 model, and experiments. Rui contributed to data preprocessing, BagNet-33 model, and model training. Yuan conducted literature review and assisted with testing and evaluation of ResNet and BagNet models. All members contributed equally to the final report. Special thanks to Dr. Avner May and Dr. Jared Dunnmon for introducing the project topic and occasional guidance throughout the workflow. The code of this project can be found at https://github.com/rui-yan/CS229-final-project. We acknowledge generous support from the Google Cloud Platform.

References

[1] He, K., Zhang, X., Ren, S. and Sun, J. Deep Residual Learning for Image Recognition. *IEEE CVPR 2016 Conference Paper.* June 2016.

[2] Brendel, W., & Bethge, M. (2019). Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. *ICLR 2019 Conference Paper.* Mar 2019.

[3] García-Martín, E., Rodrigues, C. F., Riley, G., & Grahn, H. (2019). Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, *134*, 75–88. https://doi.org/10.1016/j.jpdc.2019.07.007

[4] Lipton, Z. C. (2017). The Mythos of Model Interpretability. *ArXiv:1606.03490 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1606.03490

[5] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2019). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*. https://doi.org/10.1007/s11263-019-01228-7

[6] Wei-Wen Hsu, Min Zhang, Chen, C.-H., & Wen-Chao Yang. (2016). The use of deep learning and mean shift to learn global and local processing in human visual perception. *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, https://doi.org/10.1109/SMC.2016.7844556

[7] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2015). Learning Deep Features for Discriminative Localization. *ArXiv:1512.04150 [Cs]*. Retrieved from http://arxiv.org/abs/1512.04150