CNN Image Recognition Simplification using Patch-Based Data Reduction Techniques

Project Category: Computer Vision

Jiying Zou (*jiyingz*) Rui Yan (ruiyan) Yuan Liu (linda921)

Motivation

Nowadays convolutional neural networks(CNNs) are achieving previously unseen performance in image recognition tasks, raising questions about what core mechanism capacitates them. It is known that CNNs recognize features within images by moving filters patch-by-patch through the image. Here, an image patch is a small piece of an image that acts as a container of pixels in larger form. Next, the networks then computationally simplify these layers down and make an end vote towards what the image contains. However, the traditional approach is both time-consuming and computationally-intensive, and it is hard to understand how CNNs reach their decisions due to the complex hidden dependencies within the feature extraction process. Therefore, one motivation for this project is to decompose the CNNs and see to what extent we can simplify the architecture of image classification systems without losing much of their performance. Such attempt may even lead to hints at whether CNNs operate similarly to human vision in that they have access to global object shape information.

Another motivation originates from the practical side. Normally, when we pass an image into a CNN, activation values for the full image at each level of the structure need to stored. In fact, the storage of activation values is what takes the most memory in an operating system. Therefore, if there exists a way to simplify the architecture such that there is no need to store all activation values, it would lead to a system benefit of training models using less memory, which may potentially allow us to better tackle image recognition on high resolution images like medical or satellite images.

This project is mainly built on the application of existing methods on image recognition using CNN and would not involve theoretical results.

Method

We first aim to see how simple and accurate we can make individual patch classifiers. Ideally, each patch will output a class prediction for what is in the patch. This can be seen as a mini-image recognition problem within the context of the larger one.

There are a couple of possible approaches. Recent developments have tried to simplify the architecture of image classification by approximating traditional CNN algorithms. One paper does so by counting local feature occurrences in a *spatially-independent* manner and then aggregating counts to make a final prediction, resulting in validation performance almost comparable to those of leading deep neural networks (e.g. VGG-16) (Brendel and Bethge 2019). This bag-of-feature-based approach is known as BagNets, which we can use to produce patch class predictions. Another approach is to use linear classifiers over scale-

invariant feature transform (SIFT), which is another feature-extraction-based object-recognition algorithm for images. It compares image features to those of reference images within a database, and outputs classifications based on the most likely reference class.

After getting individual patch class predictions, we need to figure out how to aggregate them in a meaningful way as to predict the entire image. The most obvious way is to get a majority vote of all the patch classes to serve as the prediction for the entire image. However, we expect that this might fail when there is a large presence of background patches. To deal with this, we can weight patch predictions by whether they are a background patch or not, which for example can be proxied by considering the distance from the center (i.e. center patches are weighted more). We can also come up with other heuristics (simple rules) about how to aggregate the patch predictions.

Intended experiments

First, we will investigate the classification performance of multiple state-of-the-art patch-level classifiers (e.g. linear DNN-based BagNets (Brendel and Bethge, 2019), and Linear classifiers over SIFT feature and tile2vec feature), for a certain patch size. We would like to compare both the runtime and the accuracy of those classifiers. For accuracy analysis, we will use AUC-ROC as our metric. The ROC curve is a plot of the true positive rate against the false positive rate at various thresholds and the area under the curve (AUC) is equal to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. The best possible solution would yield an AUC of 1 which means we would classify all positive samples correctly without getting any false positives.

Thereafter, we will perform Gradient-weighted Class Activation Mapping (Selvaraju et al., 2017) to produce a coarse localization map highlighting the areas that the model considers important for the classification decision. This could provide us some insights via visualization on how some noisy patches such as background patches would affect the performance of patch-level classifiers. If they indeed hurt the performance of patch-level classifiers, we will consider deriving corresponding techniques to detect and ignore those noisy patches during our training process.

Once we figure out what patch-level classifier has the highest accuracy, we would like to train the patch-level classifier on our training image patches, and propose a decision fusion model to aggregate patch-level predictions for a patch-based image-level classification. We can use the same metric above, i.e. AUC-ROC, to compare multiple decision fusion models and see which one could achieve the highest accuracy.

Citations

Brendel, W. and Bethge, M. *Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet.* ICLR 2019 Conference Paper. https://arxiv.org/pdf/1904.00760.pdf
Ramprasaath R. Selvaraju. *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*. ICCV 2017 Conference Paper. https://arxiv.org/abs/1610.02391.pdf