Boosted Convolutional Neural Networks

Mohammad Moghimi[1]
mmoghimi@cs.cornell.edu

Mohammad Saberian[2]
esaberian@netflix.com

Jian Yang[3]
jianyang@yahoo-inc.com

Li-Jia Li[5]
lijiali@cs.stanford.edu

Nuno Vasconcelos[4]
nvasconcelos@ucsd.edu

Serge Belongie[1]
sjb344@cornell.edu

[1] Cornell Tech, New York, NY
 Cornell University, Ithaca, NY

[2] Netflix, Los Gatos, CA

[3] Yahoo Research, Sunnyvale, CA

[4] UC San Diego, La Jolla, CA

[5] Snapchat, Los Angeles, CA

In this work, we propose a new algorithm for boosting Deep Convolutional Neural Networks (BoostCNN) to combine the merits of boosting and these networks. To learn this new model, we propose a novel algorithm to incorporate boosting weights into the deep learning architecture. More specifically, in each iteration of boosting, we train a deep network to approximate the boosting weights, i.e. minimizing

$$\mathcal{L}_{se}(w, g) = \sum_{x_i \in \mathcal{D}} \sum_{j=1}^{M} (g_j(x_i) - w_j(x_i))^2,$$

where $g(x)$ is a deep network, $w(x)$ are boosting weights, M is the number of classes and $\mathcal{D}$ is the training set. Experiments show that the proposed method is able to achieve state-of-the-art performance on several fine-grained classification tasks such as bird, car, and aircraft classification, see Table (1) as an example.

In addition we also show that it is possible to use networks of different structures within the proposed boosting framework. In this case, at each boosting iterations, we train these networks independently to approximate the boosting weights and the network that leads to the best performance will be added to the ensemble. Experiment show that, this not only results in superior performance but also reduces the required manual effort for finding the right network structure, see Table (1).

Our open source implementation is based on Caffe framework and it is available on Github[1].

Method	Accuracy
BoostCNN	85.6%
BoostCNN (heterogeneous)	**86.2**%
Bilinear CNN (B-Net)	84.1%
Krause *et al.*	82.0%
Pose Normalized CNN	75.7%
Part-based RCNN	73.9%

Table 1: Performance comparison for bird classification on CUB200 dataset.

---

[1]http://github.com/mmoghimi/BoostCNN