

Measuring the Impact of AutoAugment on Classification Performance for Additional Image Datasets

Quoc N. Le, Rounak Mehta, Vikram Natarajan
Columbia University

{qn12000, rm3652, vsn2113}@columbia.edu

Abstract

The AutoAugment [1] authors show how automatically-selected policies for data augmentation can improve on benchmark classifier performance on CIFAR-10, CIFAR-100, SVHN, and ImageNet datasets. In this paper, we aim to understand the impact of the AutoAugment approach on classifiers for additional image datasets. We choose these additional datasets from publicly available datasets on the Kaggle, which provides both (A) a good variety of labeled image datasets and (B) a selection of repeatable baseline neural network classifiers in the form of Kaggle Kernels. By applying AutoAugment to a greater variety of image datasets and classifiers, we hope to better understand how well the AutoAugment generalizes to diverse image data and the range of performance increases that can be expected purely by data augmentation.

1. Introduction

AutoAugment shows the potential for automatically discovering data augmentation techniques that can be “learned” from the image dataset itself. It is a reinforcement learning algorithm which finds optimal policies for data augmentation for training deep learning algorithms. It works by setting the problem up as a discrete search problem, where each policy in the search space corresponds to five sub-policies (each having 2 operations). Reinforcement Learning is used to find optimal policies, then these policies are used to augment a dataset before training with a selected neural network architecture.

The original AutoAugment paper [1] demonstrated the performance of AutoAugment on benchmark datasets such as CIFAR-10, CIFAR-100, SVHN, and ImageNet. We aim to expand this work by evaluating the performance of AutoAugment on a number of real world datasets sourced from Kaggle competitions.

2. Previous Work

Besides the original AutoAugment paper [1] and its github implementation, there are also a number of github repositories that attempt to reproduce the results of the paper (DeepVoltaire/AutoAugment, rpmcruz/autoaugment, dndnjs/pytorch-cifar10). We found one github repository that tests AutoAugment on an additional image dataset of Urban images (deruhart/AutoAugment-Urban).

3. Datasets

We plan to run AutoAugment on Kaggle image datasets that meet the following criteria, which will allow us to start our evaluation quickly and maximize our fairly limited computational resources:

- Image dataset is labeled, i.e. classification is the goal
- Image dataset is of a manageable size (less than 10GB)
- A Kaggle Kernel is available (ensures repeatability)
- Kernel is a single model that can be run relatively quickly (under 1 hour). (Note this excludes many competition-winning models which are large ensembles and take days to run).

As of the time of writing this proposal, the Kaggle datasets identified include Statoli/C-Core Iceberg, Fruits 360, Flowers Recognition, and Dogs vs Cats.

4. Evaluation Criteria

For each dataset, we will measure:

- Performance of the classifier baseline (Kernel) without any data augmentation. This will require removing and data augmentation added by the Kernel author, if any.
- Performance of the classifier with data augmentation added by the Kernel author.
- Performance of the classifier with data augmentation policies discovered by AutoAugment.

The definition of "Performance" will vary for each dataset as each Kaggle competition. Some Performance measures may include Macro F1 Score or Cross Entropy Log Loss.

5. Milestones

- Select Kaggle datasets and kernel.
- Using the original paper [1] implementation in github, create AutoAugment "data loaders" for each dataset. This is needed for AutoAugment to find optimal augmentation policies for the loaded datasets.
- Run performance evaluation as described in Evaluation Criteria section.
- Analyze results and draw conclusions about AutoAugment effectiveness on chosen datasets.

6. References

[1] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation policies from data, 2018.