TINE-CNN Augmentation:
Automatic data augmentation for any image classification task

Robbie Ostrow – ostrow@cs.stanford.edu

Summary

We introduce a new data augmentation technique for image classification problems. The method, TINE-CNN augmentation, requires no human knowledge about the classification task and can effectively augment a dataset without supervision.

TINE-CNN Augmentation?

Training **I**mages **N**aively **E**mbedded in **C**onvolutional **N**eural **N**etworks

TINE-CNN Architecture

Input (32x32x3 or 64x64x3)
3x3 Convolution (16 filters)
3x3 Convolution (16 filters)
2x2 Max Pool Dropout (in training, drop 80%)
FC 256
Dropout (drop 80%)
FC 2 (binary classification)

The Algorithm

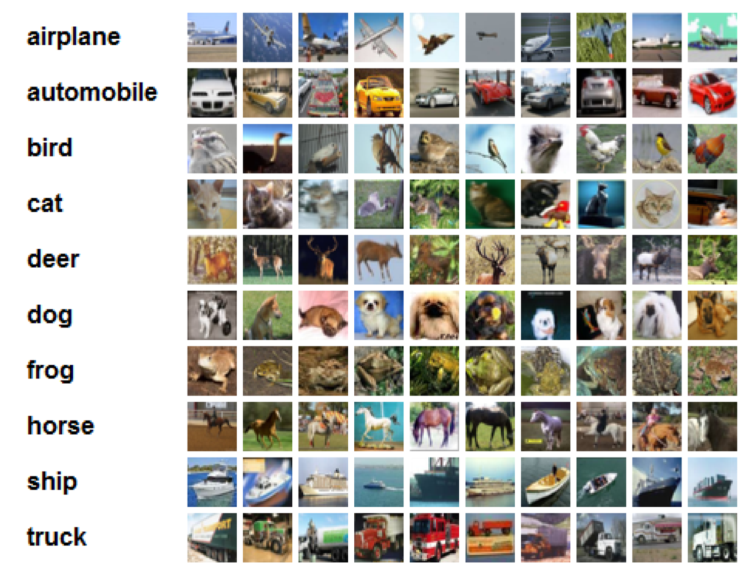
- Train 3 TINE-CNNs per class. "True" inputs are the class members, and "false" inputs are random samples from the rest of the training dataset.
- Generate training data:
  - Randomly transform elements of the training set. Use an aggressive technique, like kitchen-sink augmentation.
  - If the median classifier thinks the transform is a class member with sufficient probability, accept the sample. Otherwise, pass on the un-transformed element.
- We can generate data indefinitely using any augmentation technique; not just kitchen-sink augmentation!

Experimental setup

All experiments were run on a Google Cloud machine with 2 NVIDIA Tesla K80 GPUs, 16 vCPUs, and 104 GB memory.

Datasets

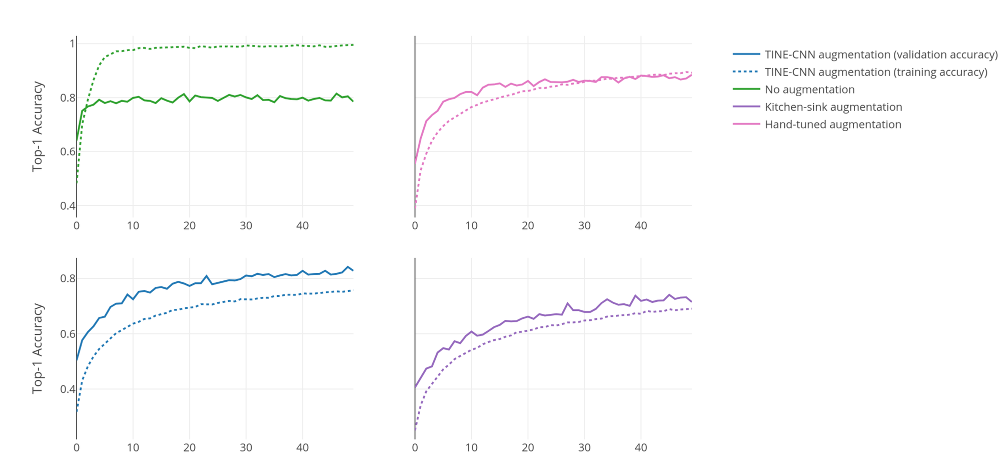
CIFAR-10 (image size: 32x32x3)
10 classes · 49,000 training images· 1,000 validation · 10,000 test

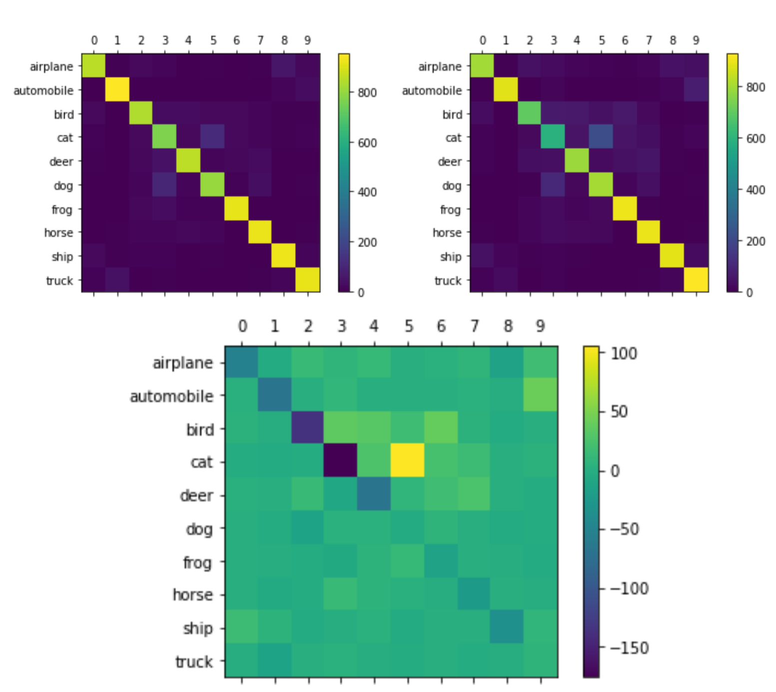


Samples from CIFAR-10 (via Andrej Karpathy)

TinyImageNet (image size: 64x64x3)
200 classes · 90,000 training images· 10,000 validation · 10,000 test

CIFAR-10 Results

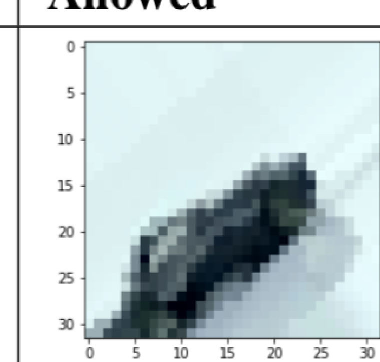
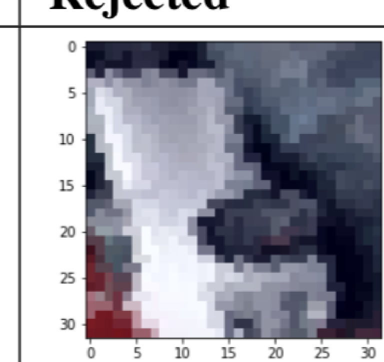
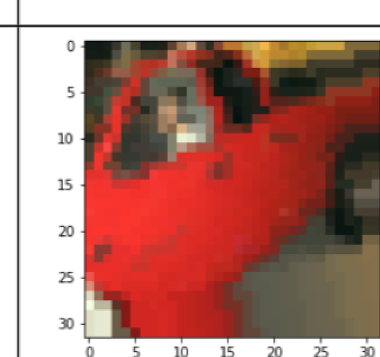
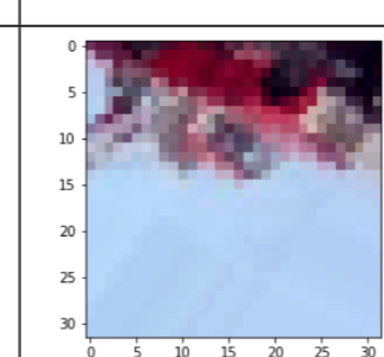
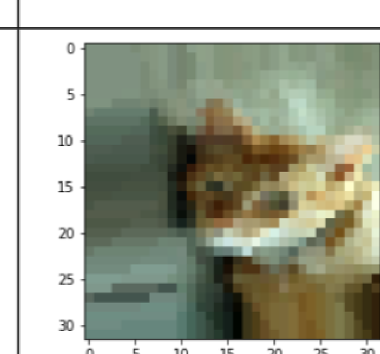
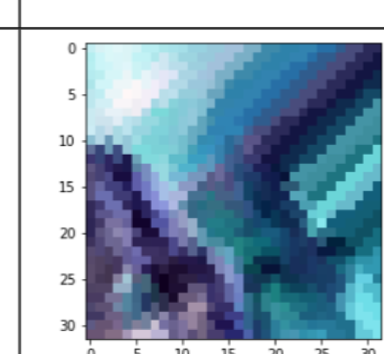




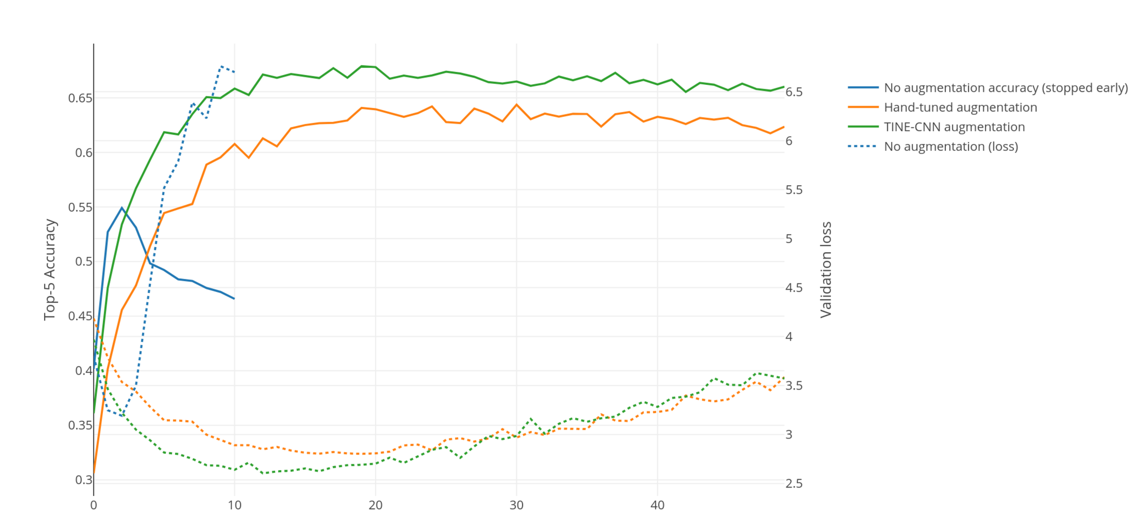
Confusion matrix for hand-tuned (left) and TINY-CNN augmentation(right). Difference (center) shows that TINY-CNN augmentation helps differentiate between CAT and DOG classes.

Types of Augmentation

- **No augmentation** - train on the data set without modification
- **Kitchen-sink augmentation** – randomly apply zero or more transforms to each image, every epoch (random flips, zooms, rotations, color shifts, crops, etc.)
- **Hand-tuned augmentation** – choose a good set of universal augmentations based on human knowledge about the dataset.
- **TINE-CNN augmentation** – filter the data generated by kitchen-sink augmentation based on "TINE-CNNs" as described on the left.

	Allowed	**Rejected**
Automobile?		
Truck?		
Cat?		

TinyImageNet results



Evaluation

Train an identical VGG-9 model using cross-entropy loss on each dataset using each type of augmentation.

Input (32x32x3 or 64x64x3)
3x3 Convolution (64 filters)
3x3 Convolution (64 filters)
2x2 Max Pool
3x3 Convolution (128 filters)
3x3 Convolution (128 filters)
2x2 Max Pool
3x3 Convolution (256 filters)
3x3 Convolution (256 filters)
2x2 Max Pool
Dropout (in training, drop 90%)
FC 1024
Dropout (drop 90%)
FC 1024
Dropout (drop 90%)
FC (10 or 200 classes)

Conclusion

TINE-CNN augmentation performs near the same level or better than hand-tuned augmentation for both CIFAR-10 and TinyImageNet. It has the disadvantage of requiring a bit of extra preprocessing time, but the significant advantage of not requiring any ad-hoc augmentation that requires human knowledge. For any problem with many classes or when an expert is unavailable, TINE-CNN augmentation should just work – and can be dropped into any image classification task without configuration.

Critical additional information

TINE-CNN Augmentation is pronounced Tiny-CNN Augmentation.