ResNet-based Food Recognition (CIS519 Project)

Wudao Ling, Siyu Zheng, Zeshen Liu

Introduction

Convolutional neural network is one of the key driving forces in Computer Vision. We applied CNN with transfer learning ideas on food image classification. The best model on ETHZ-Food-101 dataset can achieve 77.25% Top1 and 92.90% Top5 testing accuracy, the training only takes 5 hour and 9 epochs.

Dataset and Preprocessing

Dataset

- ETHZ-Food-101

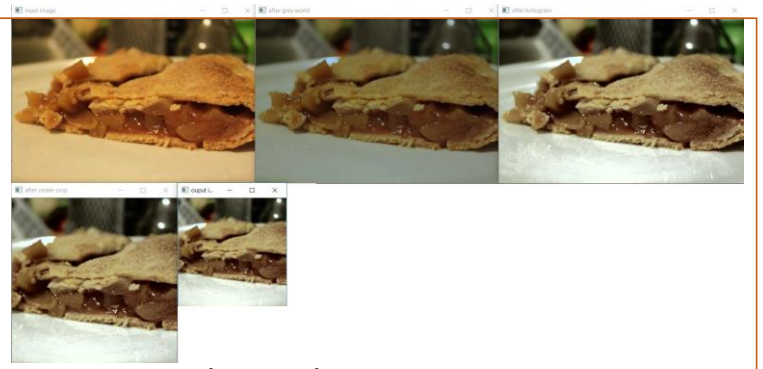


Image Preprocessing

- Center crop and resize to 224 * 224 * 3
- Implement Grey World method to normalize grey value
- Use Histogram Equalization algorithm to normalize contrast and luminance

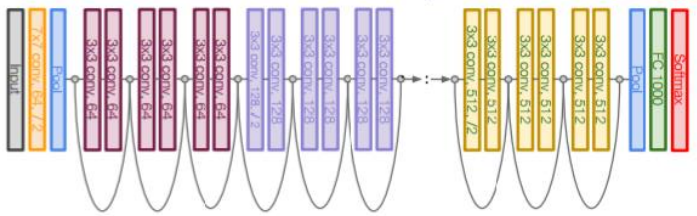
Methodology

Hardware

- Last Layer training on NVIDIA GTX 1050 GPU with 4GB memory and 16GB RAM (XPS15)
- Full network training on NVIDIA Tesla K80 GPU with 12GB memory and a 61GB RAM (AWS P2.xlarge instance)

Models: ResNet50

- 50 layer Residual Neural Network
- Adding the output f(x) derived from some residual block of conv-relu-conv series to the original input x



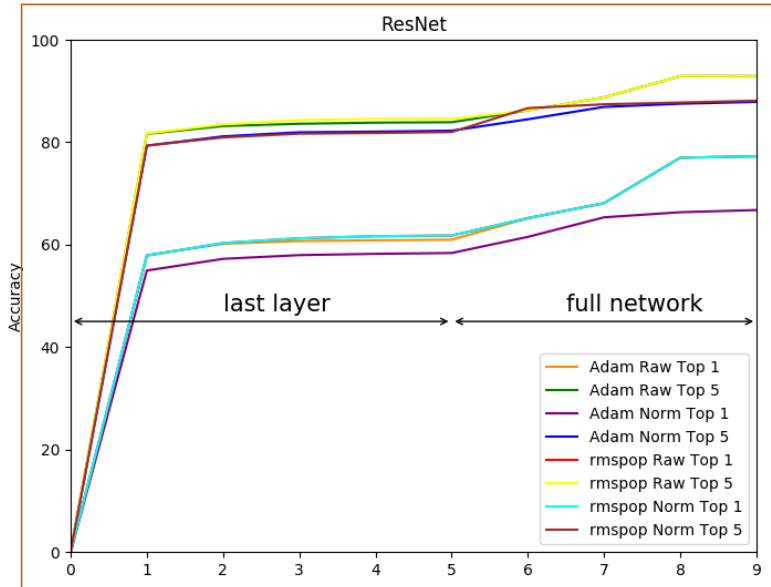
Transfer Learning

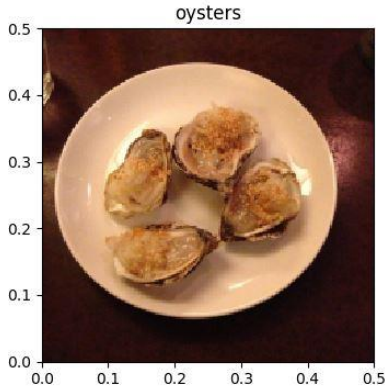
- Using ConvNet as fixed feature extractor, replacing the last dense layer and only update it for 5 epochs (~1 hour)
- Fine tuning weights of the whole pre-trained network for another 4 epochs (~4 hour)

ResNet-based Food Recognition (CIS519 Project)

Wudao Ling, Siyu Zheng, Zeshen Liu

Result





	category	probability
1	oysters	0.854446
2	baklava	0.0762466
3	crab_cakes	0.0152259
4	clam_chowder	0.0150807
5	garlic_bread	0.00583024

Analysis

Training Raw/Normalized data with RMSProp after 9 epochs

Dataset	Top1	Top5
Raw	0.7725	0.9290
Nomalized	0.6721	0.8811

Training Raw data using 2 optimizers after 9 epochs

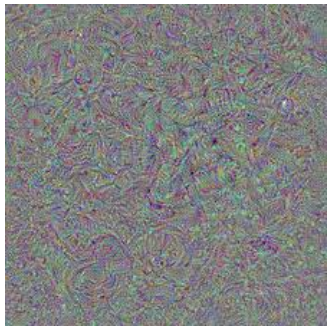
Optimizer	Top1	Top5
Adam	0.7645	0.9264
RMSProp	0.7725	0.9290

Image Preprocessing

- Images were not normalized when ResNet was trained on ImageNet, some layers may learn to understand.
- Different background environment and color contrast might be a good information for food category.

Optimizer

- RMSProp divides the learning rate by an exponentially decaying average to resolve radically diminishing learning rates. It converges faster, better in this project since training batch size is reasonably large which guarantees stability.
- Adam is essentially an RMSProp with momentum terms that dynamically adjust the learning rate using the first and second order moment estimates of the gradient. It's relatively stable.



Visualization

Generate the input image that maximizes the last layer activation of trained model, in specific categorical probability. To do that, we randomly initialize an input image and back-propagate from the last layer's output, which gives the gradient of the output w.r.t the input image pixels. Then we perform gradient ascent to get image pixels that maximize the output. This example is generated input for french_fries.