iFood-Chanllenge: Fine-grained classification of food images

Yu Wu(yw92); Yunda Jia(yj32) Team Name:comp540_yj32_yw92
Rice University

Kaggle Final Score

The final submission scored 0.11603, and place the 4th position.

Introduction

Automatic food identification can assist towards food intake monitoring to maintain a healthy diet.

Challenge: Food classification is a challenging problem due to the large number of food categories, high visual similarity between different food categories, as well as the lack of datasets that are large enough for training deep models.

Purpose: project is to find proper deep models to identify 251 fine-grained (prepared) food categories and predict the fine-grained food-category label given an image.

Data description and preprocessing

Dataset description :Training set with 251 food categories with 118,475 images, validation set of 11,994 images and the test set of 28,377 images.

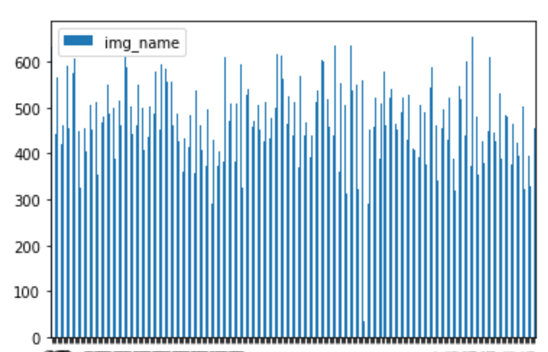
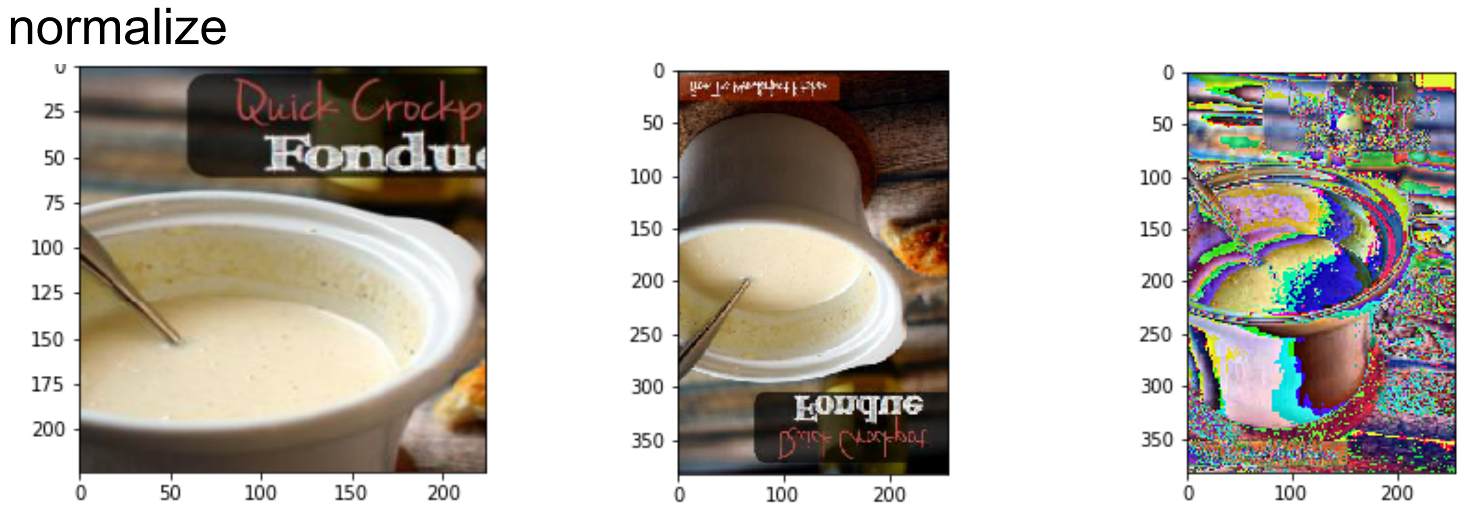


Figure 1 : Unbalanced training set

Data augmentation: Using random rotate the image from 0 degree to 30 degrees, then random crop the image into 224 * 224 pixels, random flip the image horizontally, random flip the image vertically, and normalize



(a) Random Resized Crop (b) Random Vertical Flip (c) Normalize

Figure 2: Augmentation transformation

Models and Architecture

Models
In our project, we have tried official ResNet 50, ResNet 101,ResNet 152 net, ResNext 101 net, DenseNet as our training models.

Architecture:
Two important models architecture showed bellowed, finally we choose

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

(a)ResNet Model Structure

stage	output	ResNet-50	ResNeXt-50 (32×4d)
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 128; 3×3, 128, C=32; 1×1, 256] ×3
conv3	28×28	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 256; 3×3, 256, C=32; 1×1, 512] ×4
conv4	14×14	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 512; 3×3, 512, C=32; 1×1, 1024] ×6
conv5	7×7	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 1024; 3×3, 1024, C=32; 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		$\mathbf{25.5}\times10^6$	$\mathbf{25.0}\times10^6$
FLOPs		$\mathbf{4.1}\times10^9$	$\mathbf{4.2}\times10^9$

(b) ResNeXt Structure

Figure 3: Models' Architecture

Kaggle Timeline

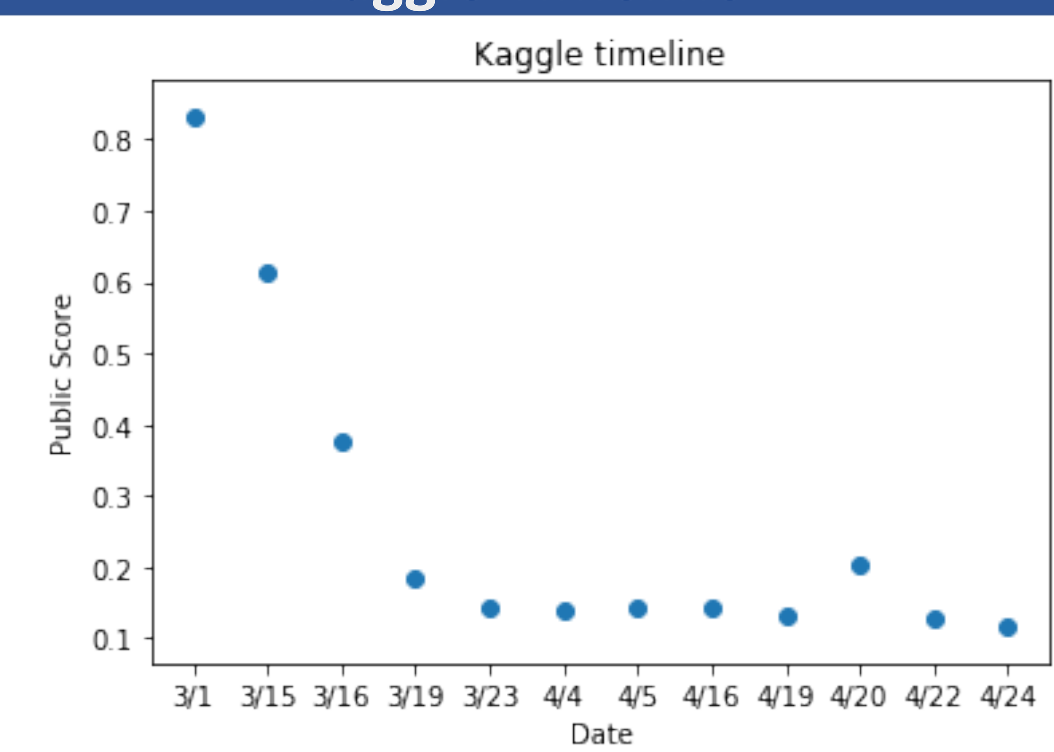
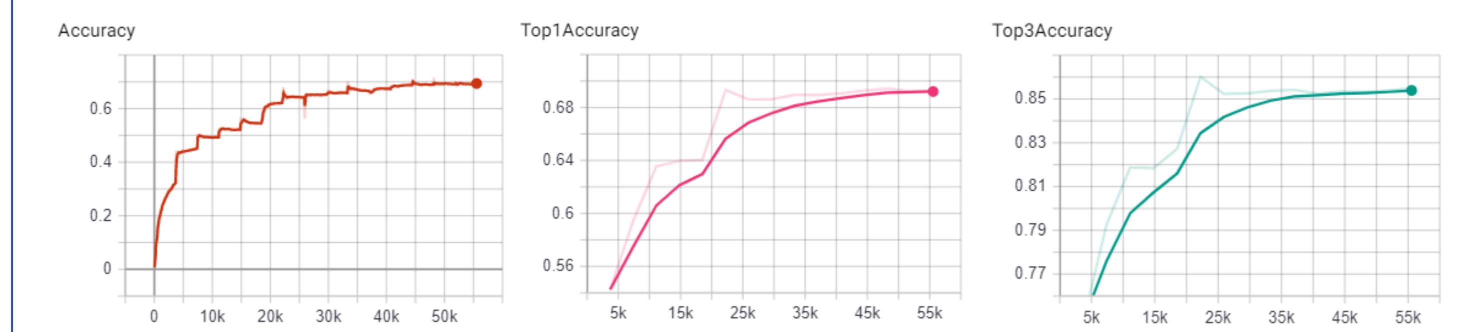


Figure 4: Kaggle submission and accuracy

Analysis and Challenges

Training platform: Use Google Collaboratory platform
Parameter analysis: Multiply base parameters for optimizer with 0.1 and remained the same fully connected layer parameters. The initial learning rate choose from 0.01, 0.001 and 0.0001 and will change after 5 epochs with gamma 0.1
Result:



(a) Training top-1 accuracy (b) Validation Top-1 Accuracy (c)Validation Top-3 Accuracy

Figure 5 : Training process of ResNeXt model

Ensemble: With different results we have, we calculate the frequency of labels each image, and take the Top-3 frequent labels as the final results. Since we have a lot of results of models. We choose the pre-result which error rate is less than 20%. Then combine them together. We got the final results which performs the best is 11.603%.

Conclusions

Model performance table:

Model name	top-3 err. (test)
ResNet 101	14.612
ResNet 152	13.995
ResNeXt 101	13.990
WideResNet 101	14.883
DesNet 161	17.819
DesNet 169	19.170
DesNet 201	18.689
Ensemble	**11.603**

We assemble the results of ResNet 152, ResNeXt 101 and DenseNet 201.We reached 11.60% error rate on test result in iFood 2019 FGVC competition

Contact

<Yu Wu>
Email: yw92@rice.edu

<Yunda Jia>
Email: yj32@rice.edu

References

[1] Jungkyu Lee, Taeryun Won, and Kiho Hong.Compounding the performance im-provements of assembled techniques in a convolutional neural network.arXiv preprintarXiv:2001.06268, 2020
[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. InProceedings of the IEEE conference on computer vision and patternrecognition, pages 770–778, 2016.
[3] Saining Xie, Ross Girshick, Piotr Doll´ar, Zhuowen Tu, and Kaiming He. Aggregated resid-ual transformations for deep neural networks. InProceedings of the IEEE conference oncomputer vision and pattern recognition, pages 1492–1500, 2017.
[4] P´eter Baranyi. Tp toolbox.https://pytorch.org/hub/facebookresearch_WSL-Images_resnext/.