Multiclass Food Classification

1st Artemii Bykov
4th year bachelor - Computer Science
Innopolis University
Innopolis, Russia
ar.bykov@innopolis.ru

2nd Maxim Salo
4th year bachelor - Computer Science
Innopolis University
Innopolis, Russia
m.salo@innopolis.ru

***Abstract*—This paper describes the model for multiclass food classification. As the base model, we use pre-trained Wide-Slice Residual Networks in cooperating with Slice branch. Moreover, we add a few linear layers to finetune out the model for our dataset**

***Index Terms*—Introduction to Computer Vision, Multiclass Classification, Supervised learning, Food**

I. Introduction

Nowadays, the number of food kinds increase day by day. With developing robots and services for food cooking and delivery auto food recognition becomes one of the main parts of service. The report describes briefly implementation of Wide-Slice Residual Network cooperating with Slice Branch Network, problems we faced and results.

II. Related Work

The main related work is Wide-Slice Residual Networks. This work introduces a new deep scheme that is designed to handle the food structure. Work exploits such a learning scheme and introduce a slice convolution block to capture the vertical food layers

III. Dataset

A. Structure

It contains images of food, organized by type of food. So for each type of food, we have about 1000 hundred images.

B. Statistic

Name of metric	Value
Number of train images	70700
Number of validation images	15150
Number of test image	15150

C. Preprocessing

1) Train images:

- Random resized crop
- Random horizontal flip
- Normalization

2) Validation images:

- Resize
- Center crop
- Normalization

3) Test images:

- Resize
- Center crop
- Normalization

IV. Implementation Model

The base idea is to take some pre-trained deep learning model, combinate with something new and finetune it with few additional linear layers on out dataset.

As a base model, we take Wide Residual Network 101-2. Then we combine it this network(based on residual learning) with slice convolution layers (Slice Branch Network) in a parallel manner [1]. So we have two branches: a residual network branch and a slice network branch with a slice convolution layer.

The residual network encodes visual representation of food images. The slice network captures the vertical food layers. The residual network implements residual learning on every few stacked layers. The slice networks exploit a slice convolution, it will learn the parameters of convolution kernel that has the same width as the input - in such way the slice network acts as a vertical layer feature extractor

And finally, in the add, we add two fully connected layers separated by dropout layers.

Link to GitHub

V. Experiments and Evaluation

A. Hyperparameters

Name of parameter	Value
Resize shape	(16, 16)
Number of epochs	10/100/50
Dataset size	0.1/0.3/1
Dropout probability	0.5
Learning rate	0.01
Learning step size	30

B. Train process

- **Batch size** - 16, when we try bigger, we got exception about CUDA out of memory (we even buy Colab Pro Subscription for $10, but batch with size 128 too big)
- **Optimizer** - SGD with learning rate equals to 0.1 and momentum equals to 0.9. Moreover, we add learning scheduler with step size equals to 30
- **Loss function** - Cross entropy loss

- Each epoch separated into two part: Train and Validation. In each epoch, current validation accuracy compares with best and current is higher the model snapshot is taken.

There are two big problems. The number of layers is huge and the size of the dataset also (about 5Gb). Moreover, we have only Colab resources and no more. That is why as a result we do not have enough time to experiment with different models and hyperparameters (1 epoch with Colab GPU takes 3-4 hours - we need at least 20-30 epochs). We took default hyperparameters for such models and try to figure out what we get based only on it.

VI. Analysis and Observations

A. 1st try

We train 30 epochs on 100% of dataset:

Name of score	**Value**
Train accuracy	50.4%
Validation accuracy	56.8%
Train accuracy	56.2%

B. 2nd try

We train 100 epochs on 40% of dataset:

Name of score	**Value**
Train accuracy	66%
Validation accuracy	73.3%
Train accuracy	72.7%

C. 3rd try

We train 50 epochs on 100% of dataset:

Name of score	**Value**
Train accuracy	83.8%
Validation accuracy	91%
Train accuracy	91%

Our aim was to perform the training process on 100 epoch, but unfortunately, Colab just kicks our session and restart the kernel. But as you can see 50 epochs on the complete dataset was enough to get high accuracy. Even on the random picture from the internet, our model makes good predictions. It is most noticeable on products with slice horizontal structure

References

[1] Niki Martinel, Gian Luca Foresti, Christian Micheloni, "Wide-Slice Residual Networks for Food Recognition, December 2016.