

Garbage Classification

Write code and debug: Deng Yushan(9.9 points) Huang Zihao(9.9 points)

Data collection and labeling, poster production: Deng Kaiyuan(9.9 points) Ding Yuhao(9.9 points) Li Hengzhi(8.5 points)

Github link: <https://github.com/IT-BillDeng/Garbage-Classification>

Task&Application

- Help people sort the garbage in their daily lives
- accelerate the speed of garbage sorting, and reduce the burden of people's garbage sorting
- While helping people sort garbage, they can also practice the concept of environmental protection.

Current mainstream methods

Data:30-40 types of garbage

Model:Based on network models such as resnet34 and resnet50

Optimizer:Mainly use operators such as SGD

Activation function:Generally relu, but part of it will be changed to the superposition of multiple functions to get better results

Input:Images containing a certain type of garbage

Output:The category of garbage in the image

Innovation

Data:Enhance the image: rotation, brightness, contrast transformation, etc., and then mark the garbage in the image.

Model:Taking faster-rcnn as the basic model, with a slight improvement on it, the image will be detected and classified afterwards, which can realize the classification of multiple types of garbage in one picture.

Optimizer:Mixed use of Adam and SGD: first use the Adam operator, when the accuracy rate slows down, use SGD directly

Output:We will output all the garbage categories in the image

Algorithm flowchart

First, we used Vgg16 as our backbone, but we made some modifications on it, removed several convolutional layers, and accelerated the training speed without affecting the accuracy, because our resources are limited. After that, after passing through the Vgg16 network, enter rpn and roi pooling, and finally classify after detecting the target. The output is the image of the bbox

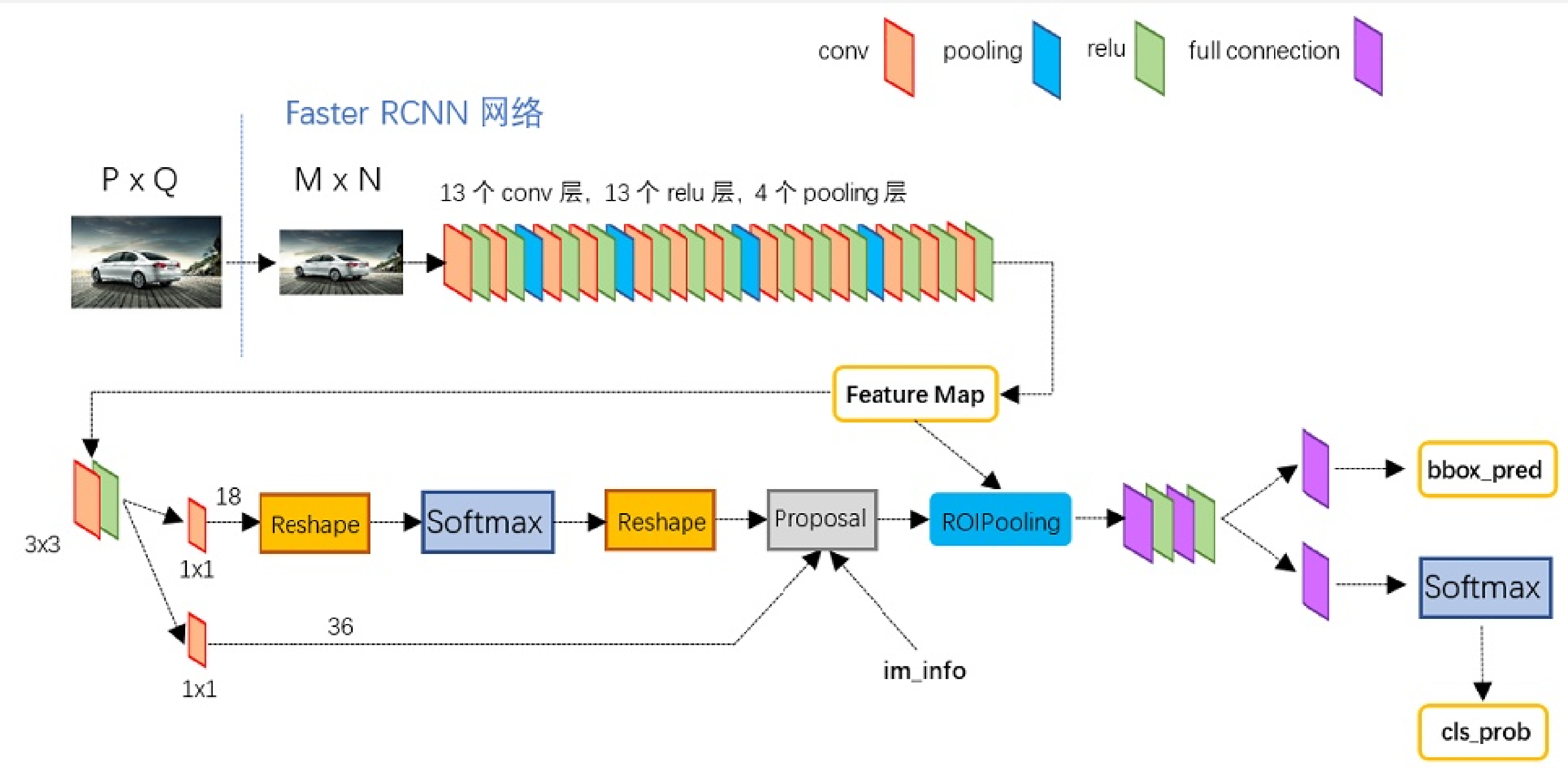


Figure: Faster-Rcnn algorithm flow chart

Results

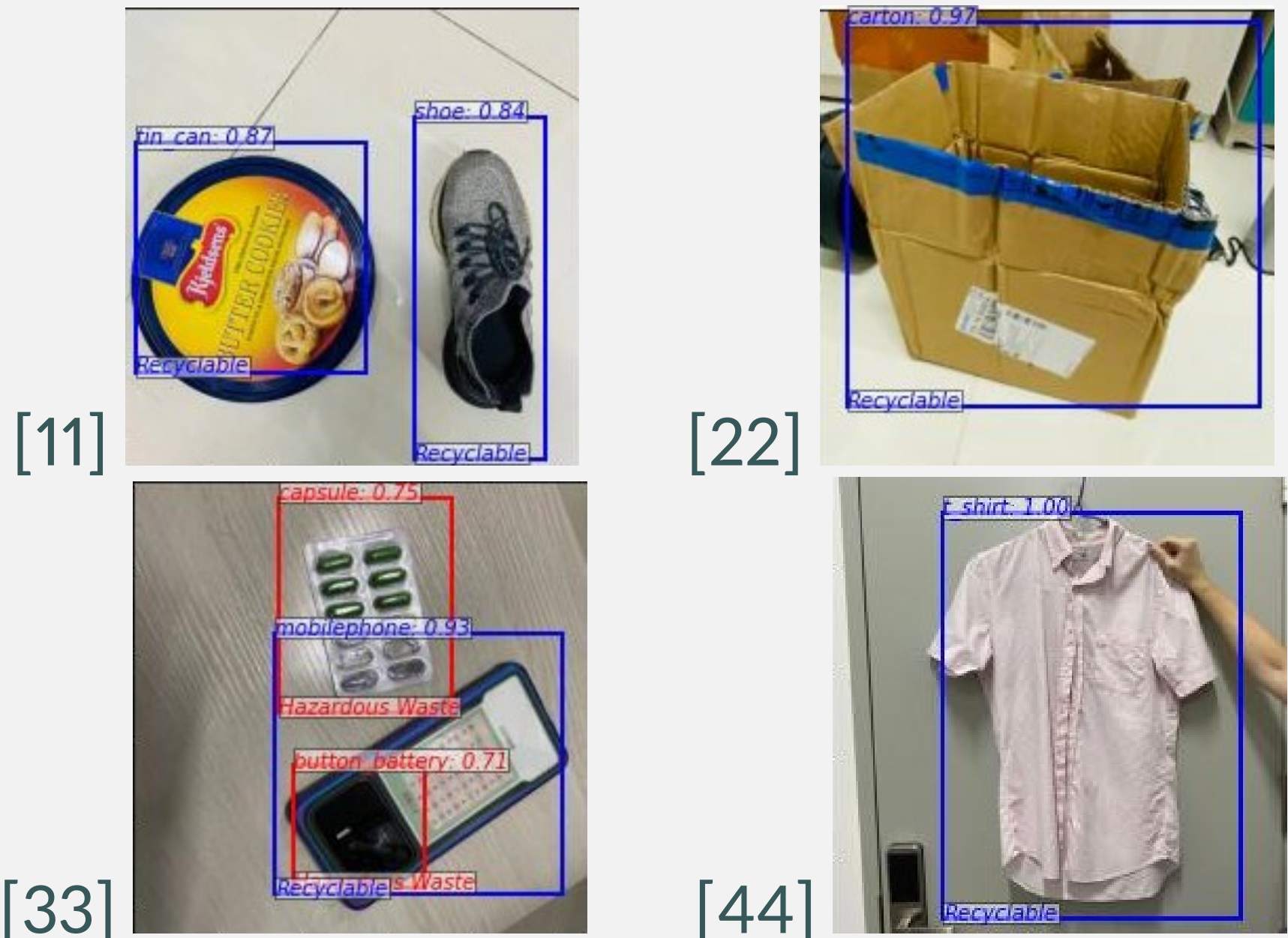


Figure: demo

Training

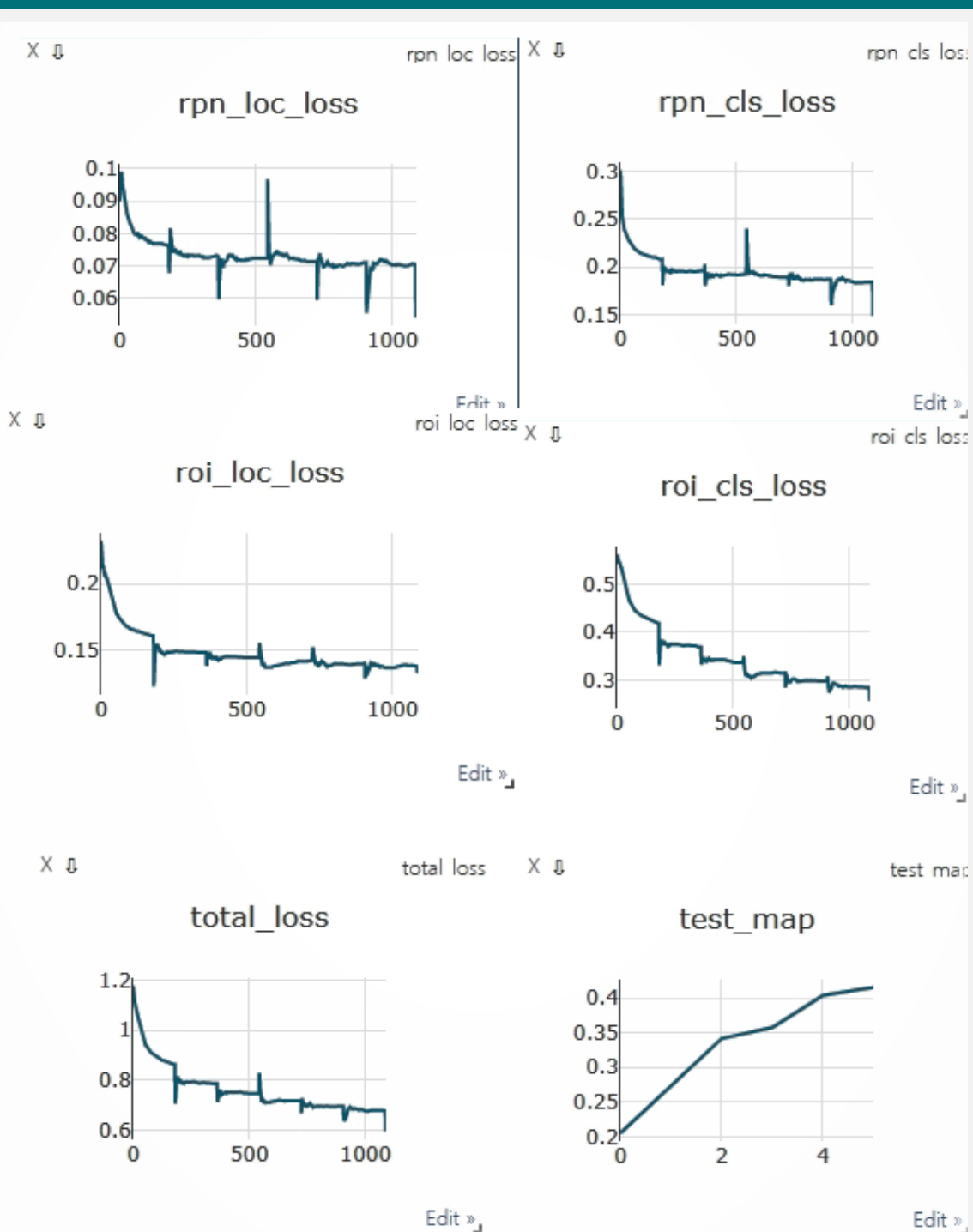


Figure: Training curve

Experimental setup and database description

GPU:Geforce RTX 2060

GPU Memory:3GB

Database:30 categories, 18,000 images with annotations

Epoch:about 60 epochs

BatchSize:1

LearningRate:0.0001

LearningRateDecay:0.01

WeightDecay:0.0005

TrainingSpeed:2.8 images/s

Self-contrast experiment

Improve	Accuracy
basicResnet50	0.80
DataEnhancement	0.88

Table: Pure classification

Improve	Accuracy
basicFasterRcnn	0.43
SimpleFasterRcnn	0.46
DataEnhancement	0.56
ChangeOptimizer	0.61

Table: Detection and classification



Figure: for fun