中国科学院大学

University Of ChineseAcademy Of Sciences

神经网络构建与地物分类

yangxue

博学笃志　格物明德

一 数据集

数据集说明：

UC Merced Land Use数据集包含21类土地类型，每类图像为100张，每张图像的像素为256*256。

数据集特点：

- 数据集比较小，每一类只有100张图片；
- 类间距离小，不同类的图片之间很相似；
- 类内距离大，同类图片之间差别较大

一 数据集

University Of ChineseAcademy Of Sciences

类间距离小：

mobilehomepark14

mediumresidential42

denseresidential35

一 数据集

类内距离大：

tenniscourt24

tenniscourt49

tenniscourt64

二 训练模型

Caffe finetune：

数据集的特点都是不利于图片的分类的，尤其是数据量太小，即使是进行了data augmentation，如果从头开始用数据集来训练网络肯定会造成严重的过拟合。考虑到这种情况，一个解决方法就是使用训练好的网络进行微调以适应我们自己的数据集，这种方法不仅能解决数据集小的问题，也能大大加快训练的速度。以下将使用Caffe深度学习框架，将数据在Caffenet模型上进行微调。

二 训练模型

实验步骤：

- 数据划分，将数据集划分为8:2；
- Data augmentation，将训练数据通过剪切、镜像等扩大八倍；
- 制作caffe要用的数据格式LMDB；
- 修改caffenet配置文件的参数（注意：微调时要修改网络名称，调大最后一层的学习率和步长，迭代次数较少）；
- 训练模型，对结果可视化分析。

二 训练模型

主要配置参数修改：

create_imagenet.sh

```
#!/usr/bin/env sh
# Create the imagenet lmdb inputs
# N.B. set the path to the imagenet train + val data dirs
set -e

EXAMPLE=/home/yangxue0827/caffe/examples/mytask
DATA=/home/yangxue0827/caffe/data/mydata
TOOLS=/home/yangxue0827/caffe/build/tools

TRAIN_DATA_ROOT=/home/yangxue0827/yangxue/UCMerced_LandUse/Train_Linux/
VAL_DATA_ROOT=/home/yangxue0827/yangxue/UCMerced_LandUse/Test_Linux/
```

Make_imagenet_mean.sh

```
#!/usr/bin/env sh
# Compute the mean image from the imagenet training lmdb
# N.B. this is available in data/ilsvrc12

EXAMPLE=/home/yangxue0827/caffe/examples/mytask
DATA=/home/yangxue0827/caffe/data/mydata
TOOLS=/home/yangxue0827/caffe/build/tools
```

二 训练模型

中国科学院大学
University Of ChineseAcademy Of Sciences

主要配置参数修改：

deploy.prototxt

```
name: "FlickrStyleCaffeNet"
layer {
  name: "data"
  type: "Input"
  top: "data"
  input_param { shape: { di
}
```

……

```
layer {
  name: "fc8_flickr"
  type: "InnerProduct"
  bottom: "fc7"
  top: "fc8_flickr"
  inner_product_param {
    num_output: 21
  }
}
```

solver.prototxt

```
net: "models/bvlc_reference_caffenet/train_val.prototxt"
test_iter: 6
test_interval: 60
base_lr: 0.001
lr_policy: "step"
gamma: 0.1
stepsize: 10000
display: 20
max_iter: 480
momentum: 0.9
weight_decay: 0.0005
snapshot: 10000
snapshot_prefix: "models/bvlc_reference_caffenet/caffenet_train"
solver_mode: CPU
```

二 训练模型

训练网络：

```
yangxue0827@ubuntu: ~/caffe
yangxue0827@ubuntu:~/caffe$ ./build/tools/caffe train -solver models/bvlc_refere
nce_caffenet/solver.prototxt -weights models/bvlc_reference_caffenet/bvlc_refere
nce_caffenet.caffemodel
```

```
I0509 20:06:24.347250  7773 solver.cpp:397]     Test net output #0: accuracy = 0.0404762
I0509 20:06:24.347414  7773 solver.cpp:397]     Test net output #1: loss = 3.21129 (* 1 = 3.21129 loss)
I0509 20:06:33.211653  7773 solver.cpp:218] Iteration 0 (2.50812e+12 iter/s, 74.805s/20 iters), loss = 3.14626
I0509 20:06:33.212383  7773 solver.cpp:237]     Train net output #0: loss = 3.14626 (* 1 = 3.14626 loss)
I0509 20:06:33.212474  7773 sgd_solver.cpp:105] Iteration 0, lr = 0.001
I0509 20:09:42.562257  7773 solver.cpp:218] Iteration 20 (0.105625 iter/s, 189.349s/20 iters), loss = 2.49532
I0509 20:09:42.562650  7773 solver.cpp:237]     Train net output #0: loss = 2.49532 (* 1 = 2.49532 loss)
I0509 20:09:42.562671  7773 sgd_solver.cpp:105] Iteration 20, lr = 0.001
I0509 20:12:54.841702  7773 solver.cpp:218] Iteration 40 (0.104016 iter/s, 192.279s/20 iters), loss = 3.0817
I0509 20:12:54.841997  7773 solver.cpp:237]     Train net output #0: loss = 3.0817 (* 1 = 3.0817 loss)
I0509 20:12:54.842017  7773 sgd_solver.cpp:105] Iteration 40, lr = 0.001
I0509 20:15:57.234267  7773 solver.cpp:330] Iteration 60, Testing net (#0)
I0509 20:16:21.667446  7777 data_layer.cpp:73] Restarting data prefetching from start.
I0509 20:17:10.487386  7773 solver.cpp:397]     Test net output #0: accuracy = 0.57619
I0509 20:17:10.487680  7773 solver.cpp:397]     Test net output #1: loss = 1.47071 (* 1 = 1.47071 loss)
I0509 20:17:20.611398  7773 solver.cpp:218] Iteration 60 (0.0752533 iter/s, 265.769s/20 iters), loss = 1.85385
I0509 20:17:20.611515  7773 solver.cpp:237]     Train net output #0: loss = 1.85385 (* 1 = 1.85385 loss)
```

二 训练模型

中国科学院大学
University Of ChineseAcademy Of Sciences

测试数据：

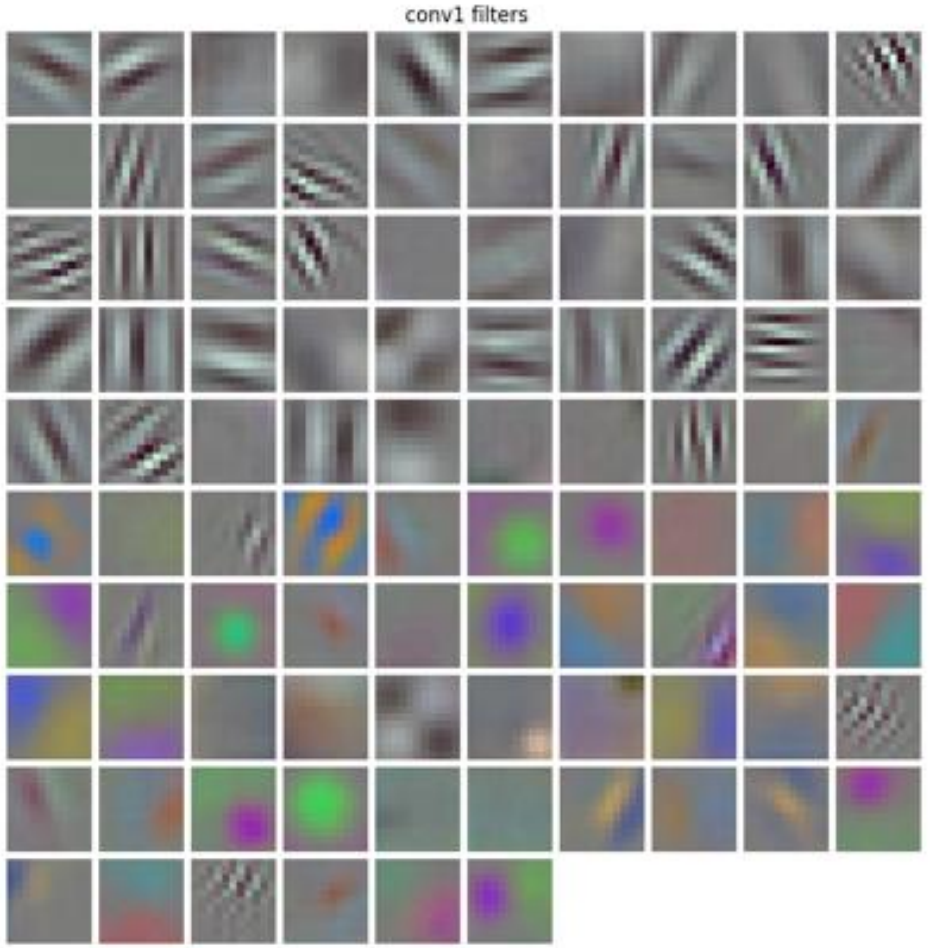
```
yangxue0827@ubuntu: ~/caffe
yangxue0827@ubuntu:~/caffe$ ./build/tools/caffe.bin test -model models/bvlc_refe
rence_caffenet/train_val.prototxt -weights models/bvlc_reference_caffenet/caffen
et_train_iter_285.caffemodel -iterations 6
```

```
I0509 20:38:52.709347  7939 caffe.cpp:290] Running for 6 iterations.
I0509 20:39:08.760025  7939 caffe.cpp:313] Batch 0, accuracy = 0.942857
I0509 20:39:08.760150  7939 caffe.cpp:313] Batch 0, loss = 0.179001
I0509 20:39:28.993718  7939 caffe.cpp:313] Batch 1, accuracy = 0.942857
I0509 20:39:28.993896  7939 caffe.cpp:313] Batch 1, loss = 0.311732
I0509 20:39:29.066073  7942 data_layer.cpp:73] Restarting data prefetching from start.
I0509 20:39:54.230525  7939 caffe.cpp:313] Batch 2, accuracy = 0.9
I0509 20:39:54.231024  7939 caffe.cpp:313] Batch 2, loss = 0.361534
I0509 20:40:11.842226  7939 caffe.cpp:313] Batch 3, accuracy = 0.9
I0509 20:40:11.842489  7939 caffe.cpp:313] Batch 3, loss = 0.49772
I0509 20:40:32.399786  7939 caffe.cpp:313] Batch 4, accuracy = 0.928571
I0509 20:40:32.399902  7939 caffe.cpp:313] Batch 4, loss = 0.276551
I0509 20:40:56.668782  7939 caffe.cpp:313] Batch 5, accuracy = 0.971429
I0509 20:40:56.670830  7939 caffe.cpp:313] Batch 5, loss = 0.081635
I0509 20:40:56.670908  7939 caffe.cpp:318] Loss: 0.284696
I0509 20:40:56.670948  7939 caffe.cpp:330] accuracy = 0.930952
I0509 20:40:56.671017  7939 caffe.cpp:330] loss = 0.284696 (* 1 = 0.284696 loss)
```

三 可视化

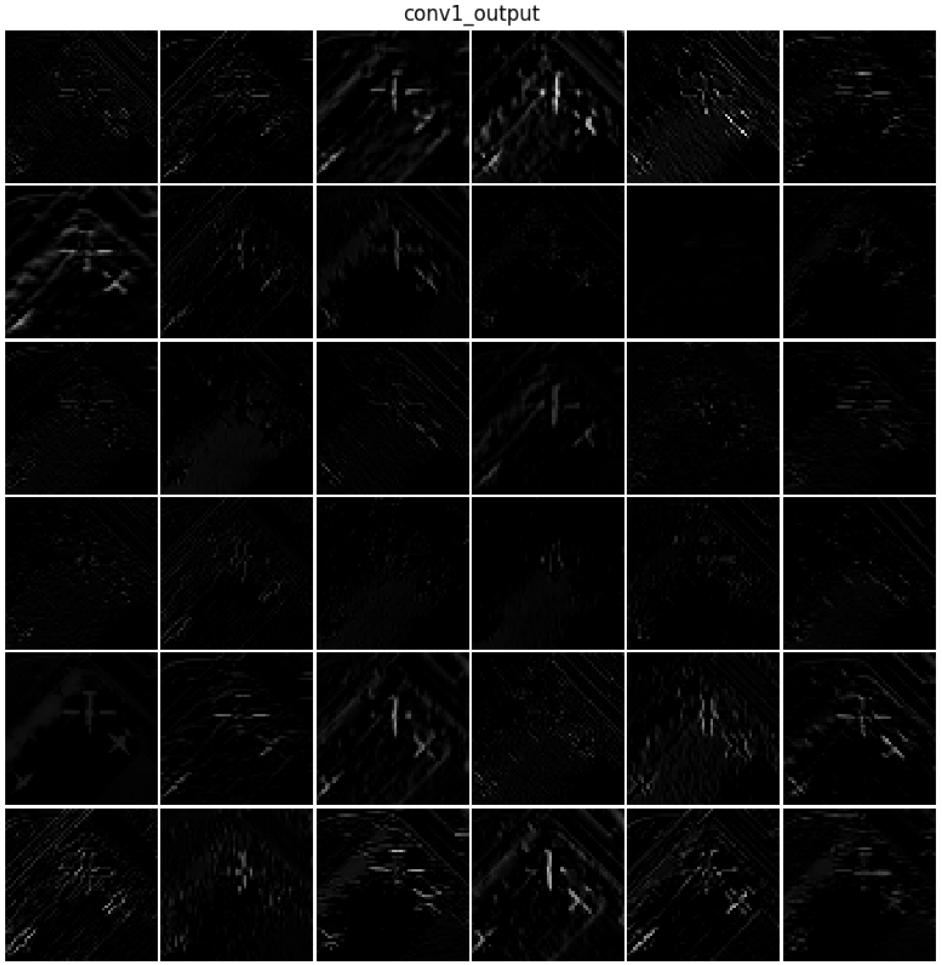
可视化网络：

caffenet网络的第一层的卷积层中我们定义了96个滤波器，96个滤波器可视化如下图所示，学过图像处理的同学都知道，下图中第一个滤波器是提取斜向下的边缘特征，第二个滤波器是提取斜向上的边缘特征，前面的滤波器大多数是在提取边缘特征，后面的大多是在统计颜色特征。



三 可视化

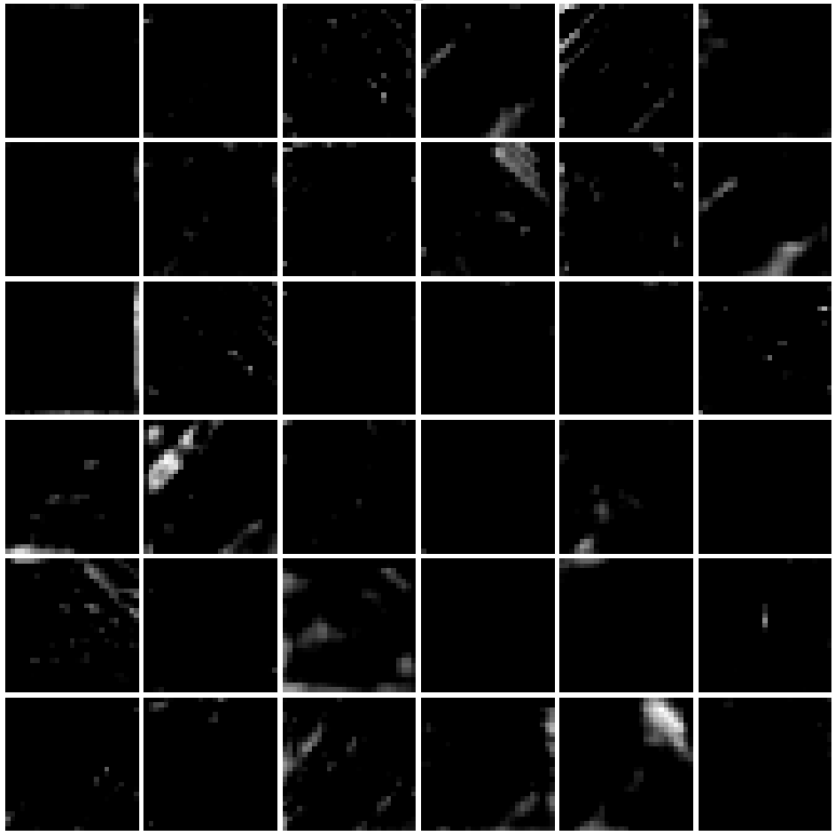
可视化网络：



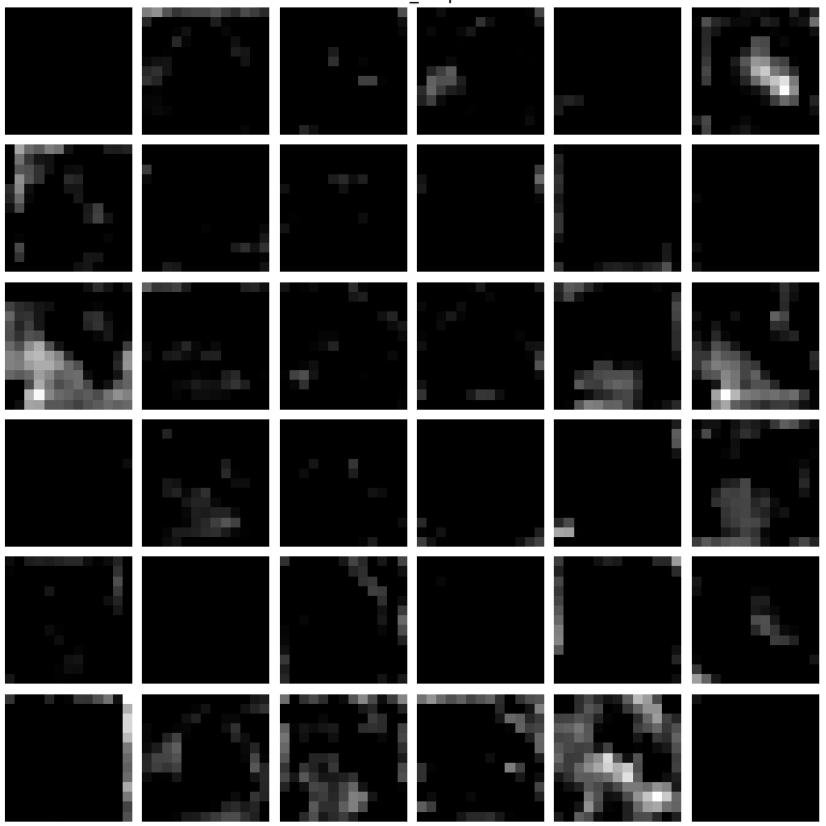
三 可视化

可视化网络：

conv2_output



conv3_output

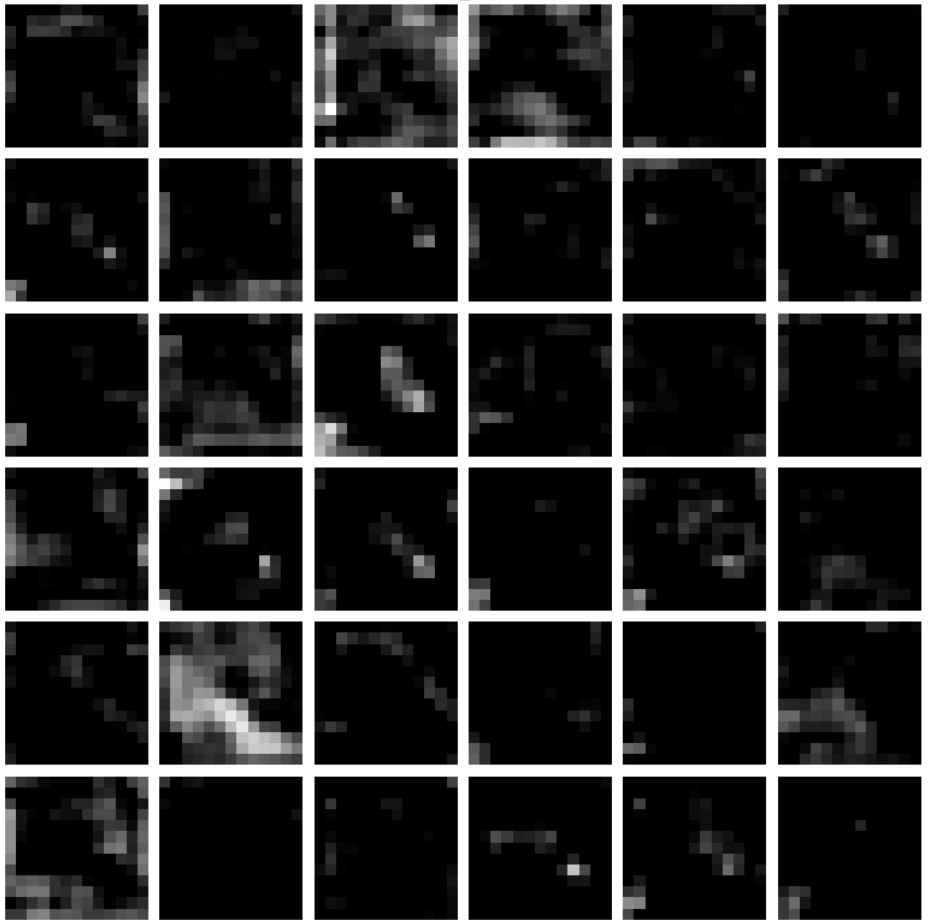


三 可视化

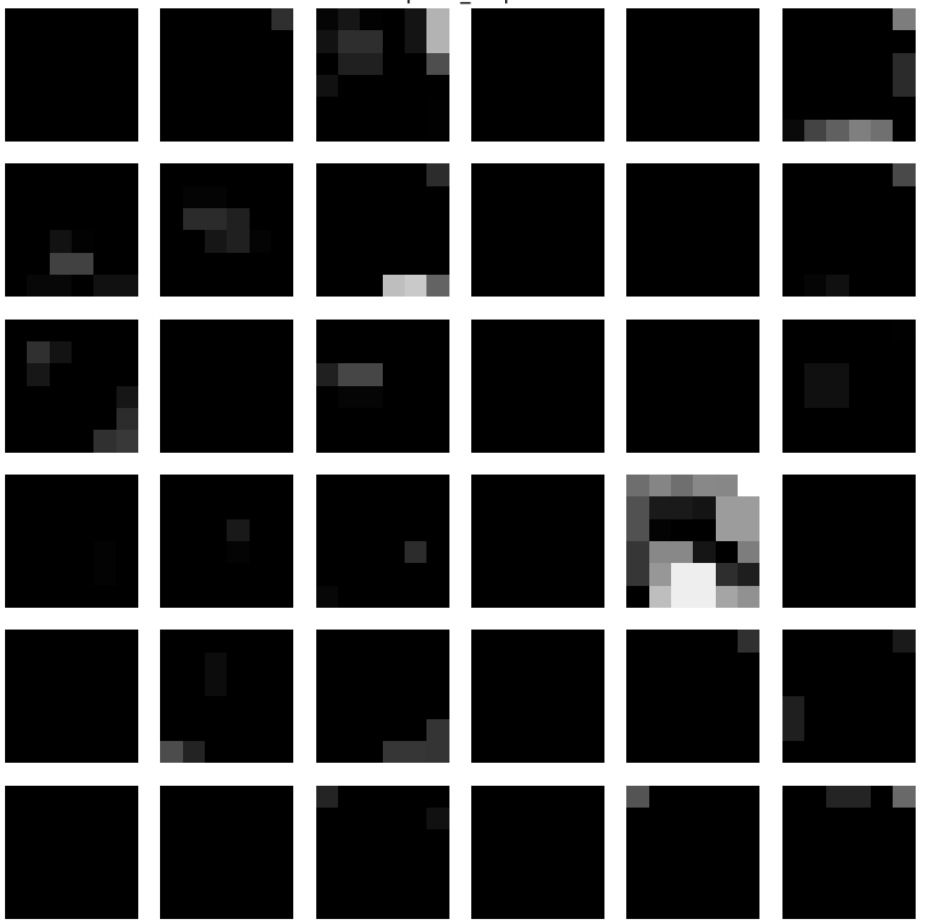
University Of ChineseAcademy Of Sciences

可视化网络：

conv4_output



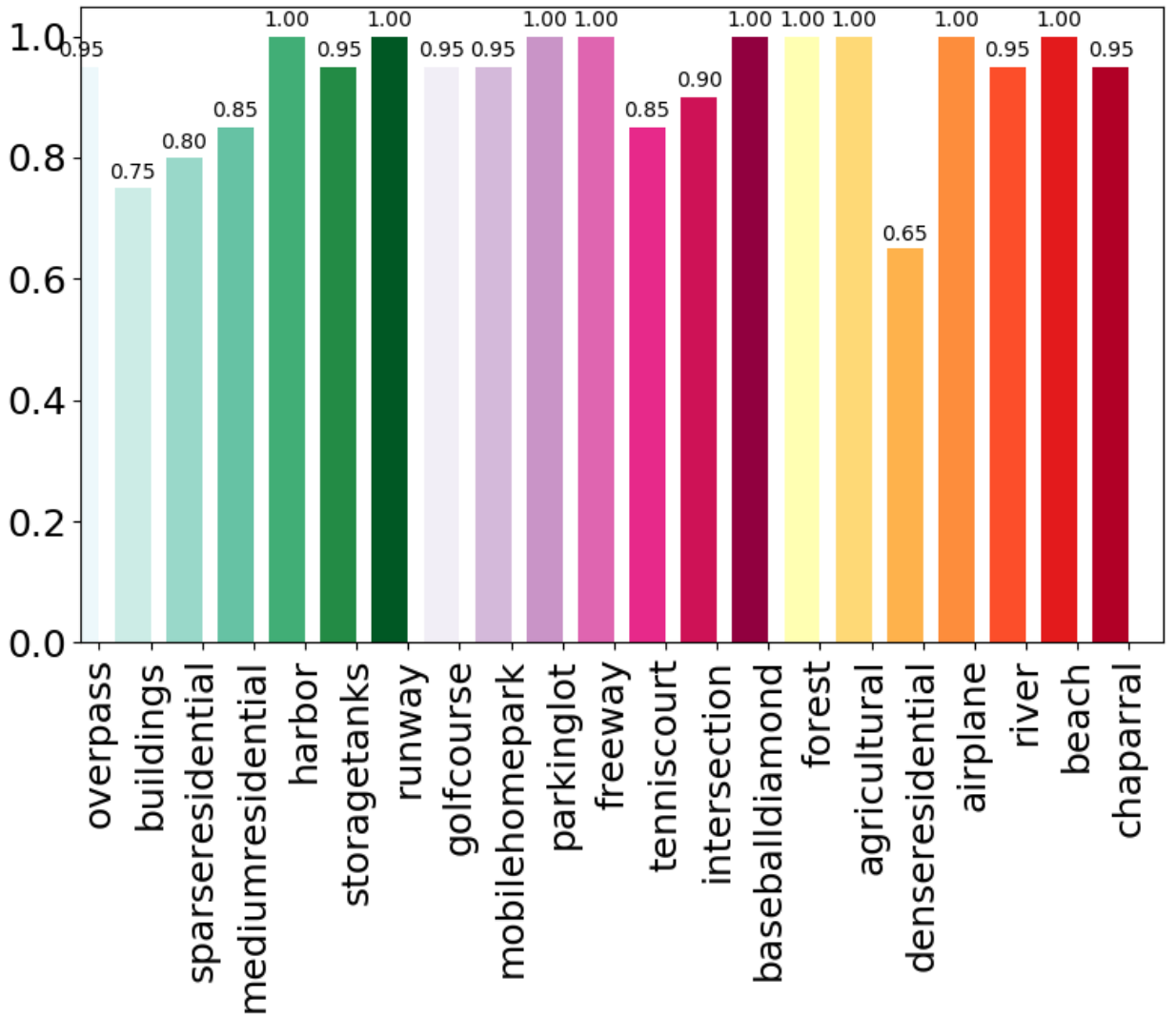
pool5_output



三 可视化

可视化网络：

从第一层到第五层卷积层滤波器输出可以看出，随着层数越高，滤波器输出比较抽象，这充分体现了卷积神经网络强大的特征提取能力。右图是每一类的分类正确率，最终测试正确率为93.1%。

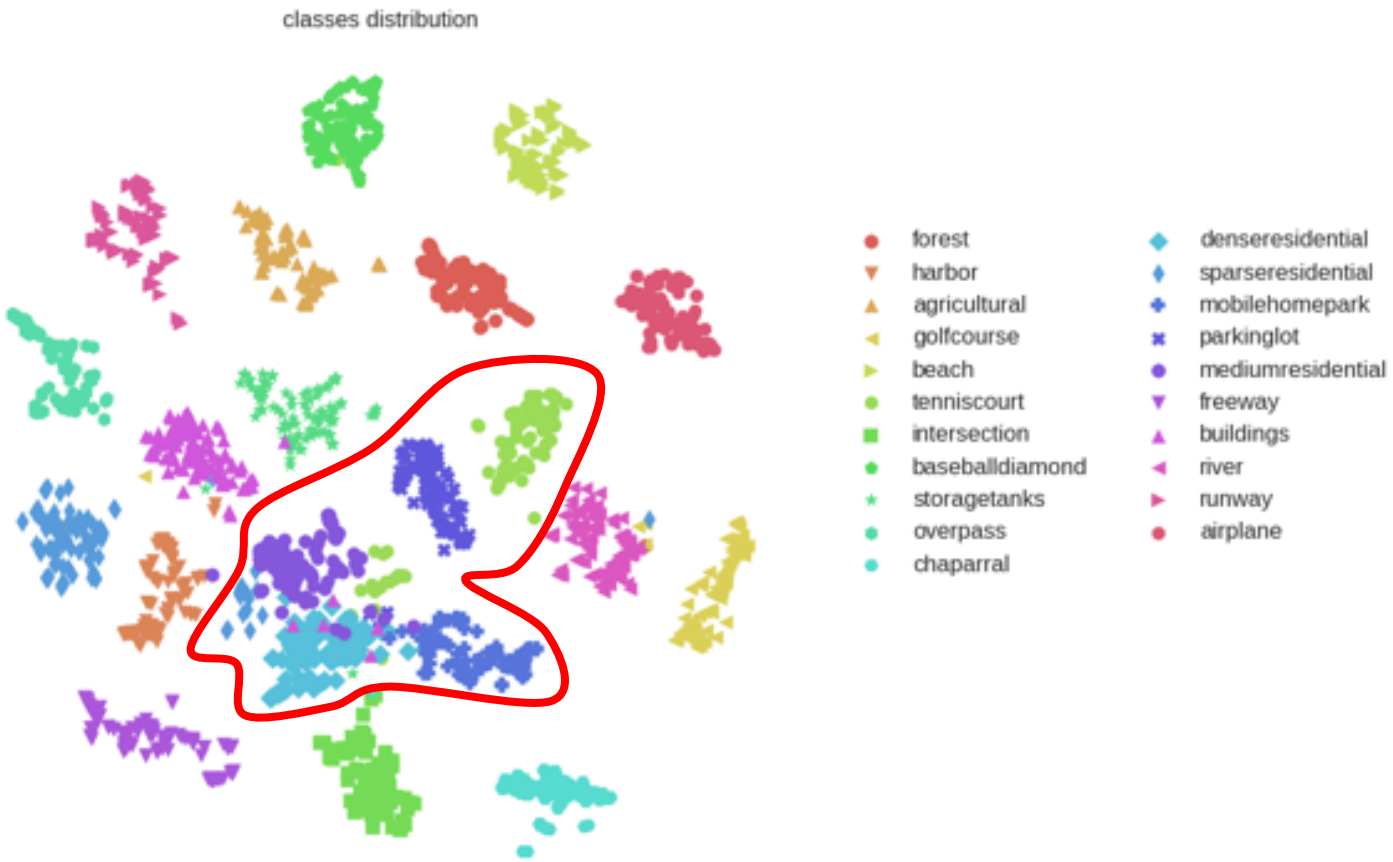


三 可视化

University Of ChineseAcademy Of Sciences

可视化网络：

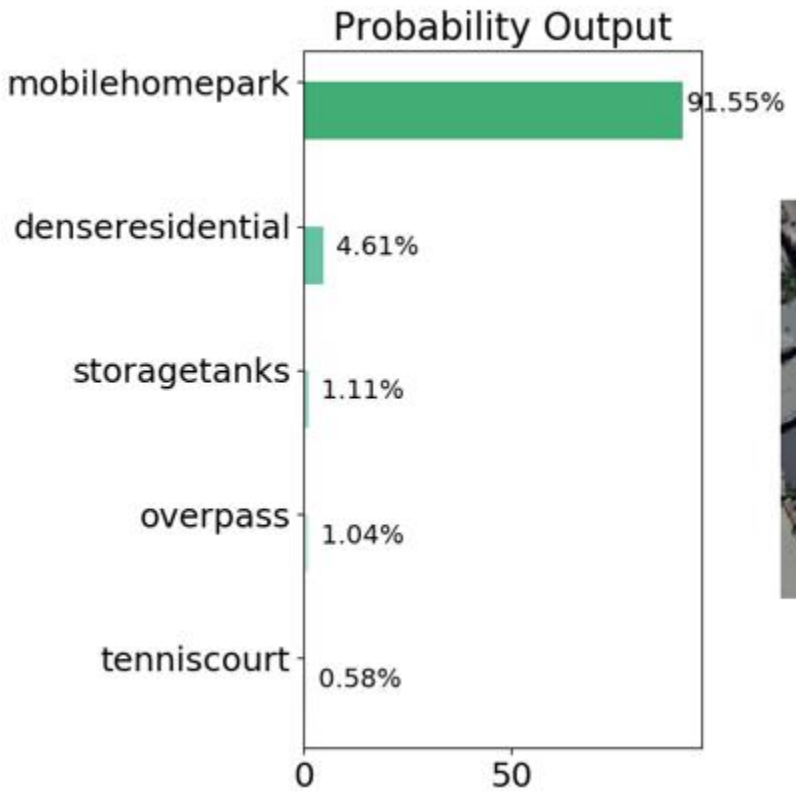
右图是每一类在网络fc7层提取4096维特征降维后的分布图，可以看出denseresidential、mediumresidential、mobilehomepark、parkinglot、tenniscourt五类较难区分。

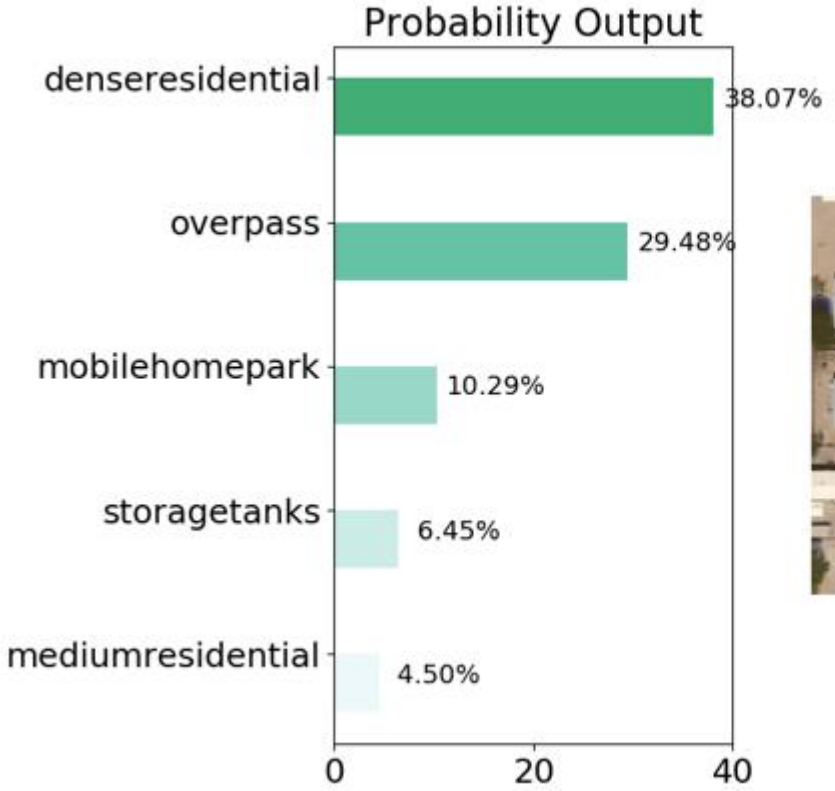


三 可视化

University Of ChineseAcademy Of Sciences

可视化网络：





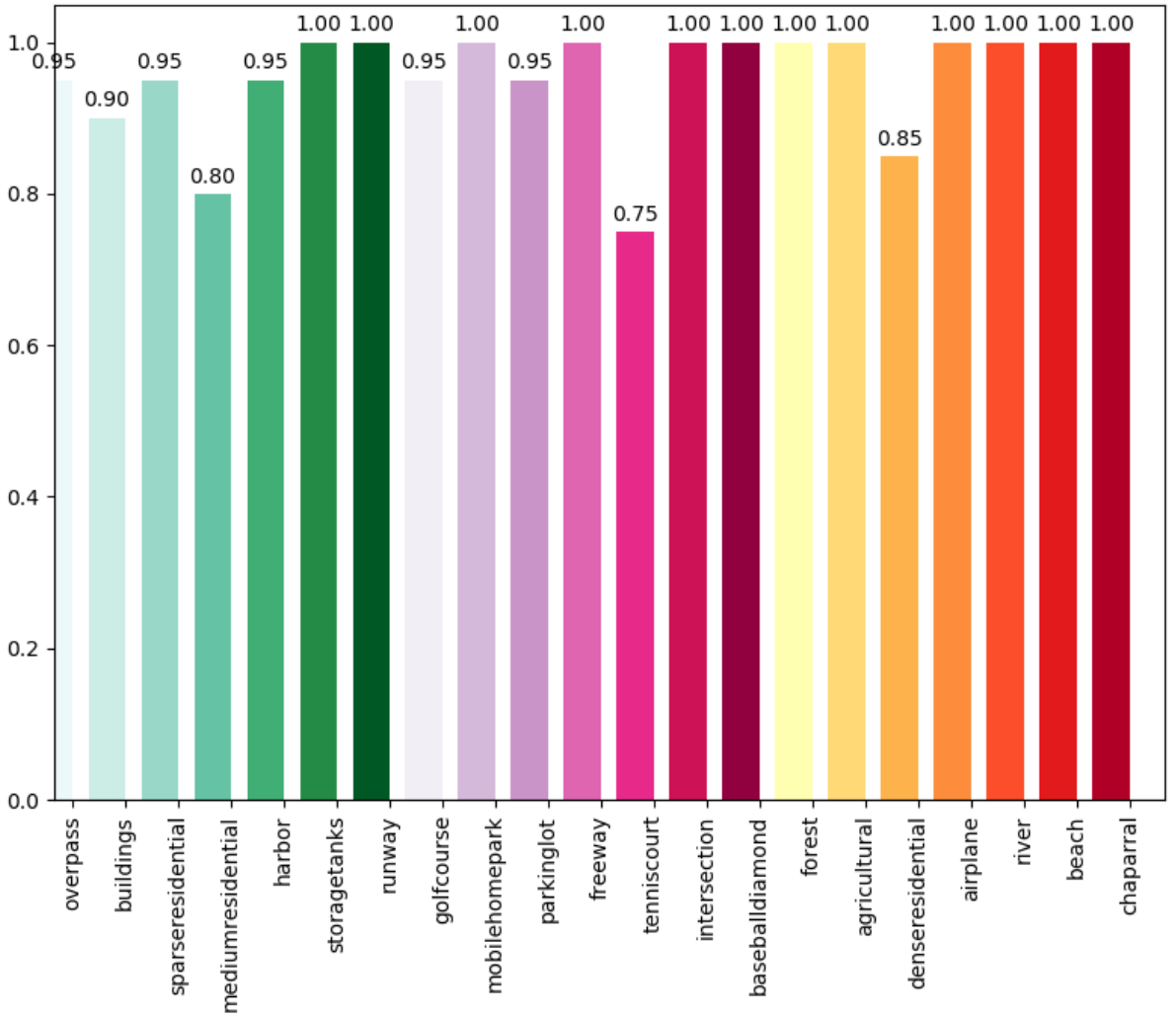
Caffe finetune + SVM：

有一种常用的方法是不用CNN的最后一层分类，用CNN提取到的特征用SVM来分类，也能达到不错的效果。在这里我们提取fc7层输出的特征，根据上面定义的网络结构，fc7层共有4096个神经元，所以每张图片的特征维数为4096维，维数比较大。

三 可视化

可视化网络：

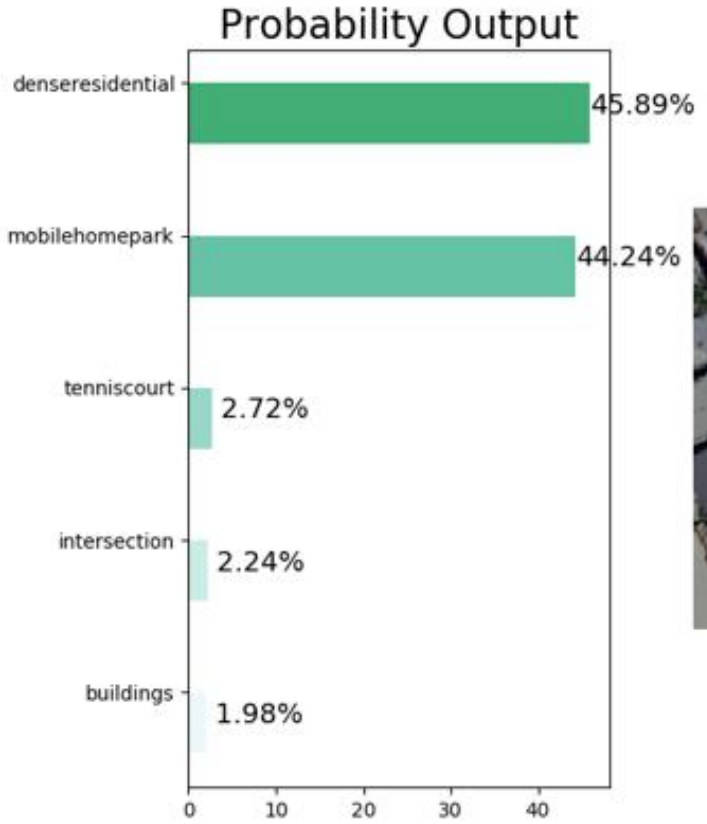
采用SVM来进行最后的分类比起网络直接分类取得了更好的效果，分类正确率达到95.2%

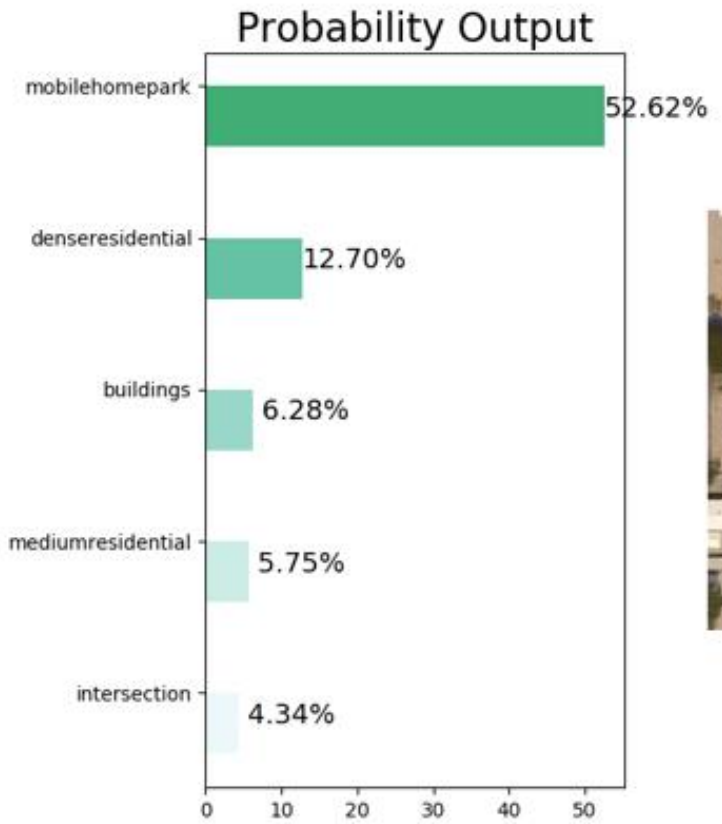


三 可视化

University Of ChineseAcademy Of Sciences

可视化网络：





中国科学院大学

University Of ChineseAcademy Of Sciences

Thank you!

博学笃志　格物明德