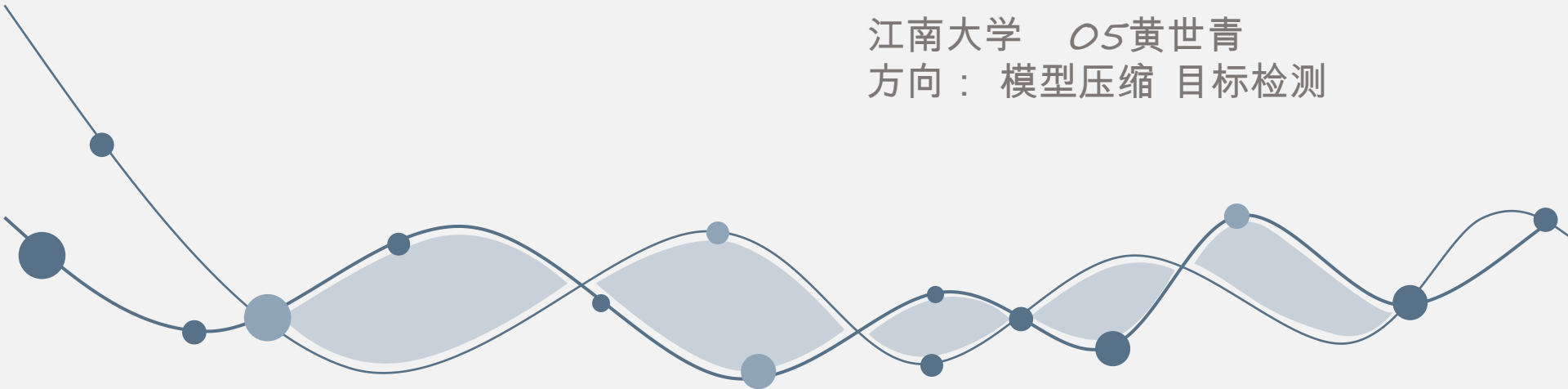
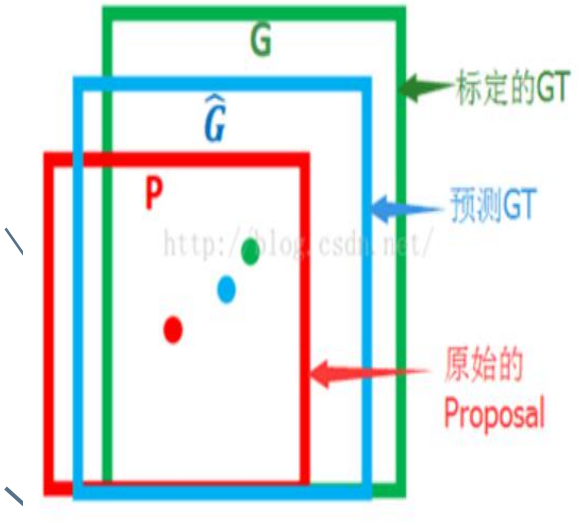
G-CNN

江南大学　05黄世青
方向：模型压缩 目标检测



效果演示

主要创新点和思路

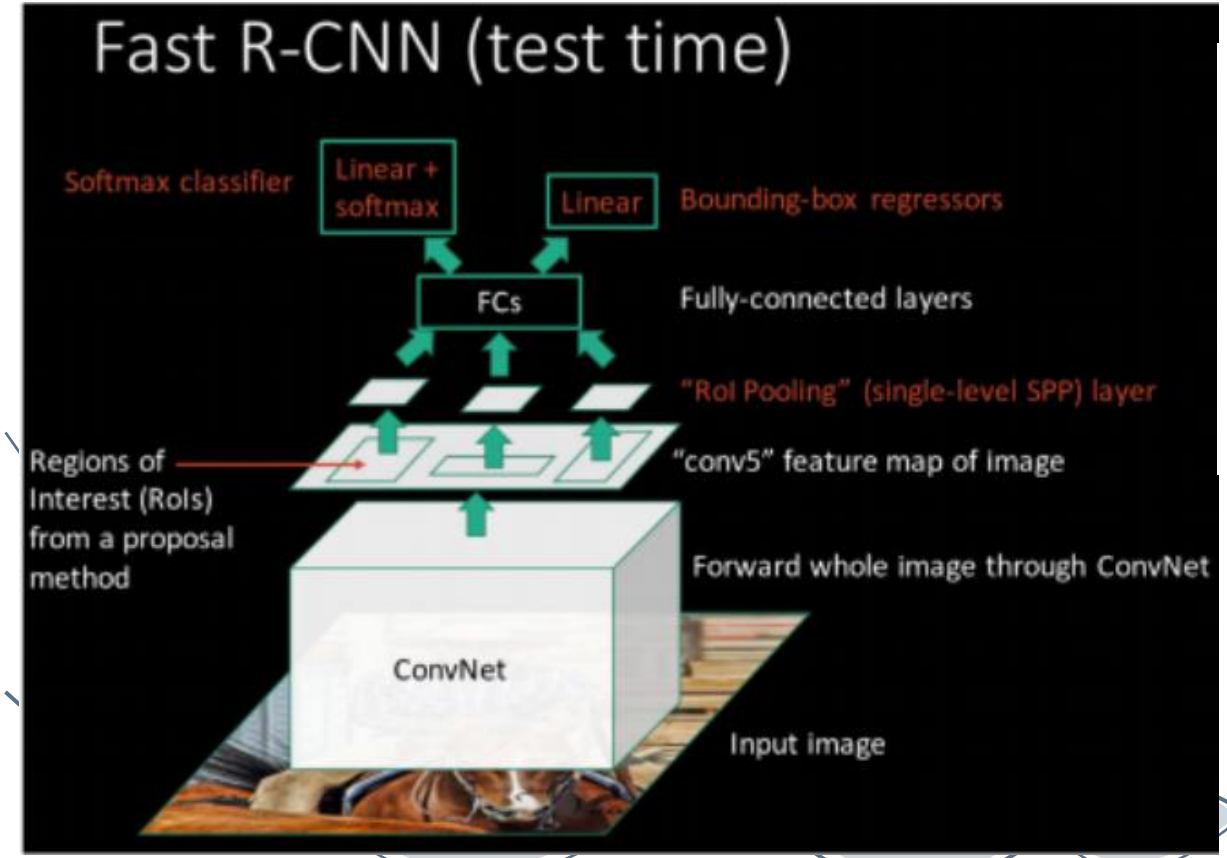


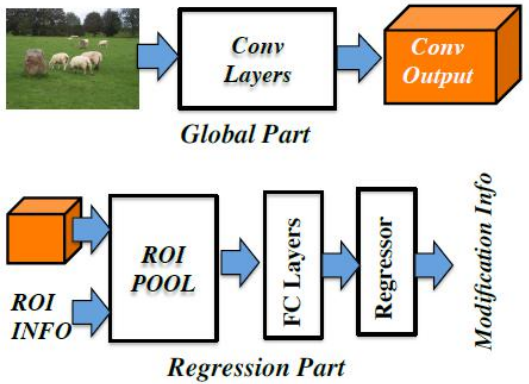
主要创新点：1.替换了Fast-RCNN的SS算法
2.将目标检测问题模型化为迭代回归问题

主要思路：直接回归最终位置比较难，但是可以每次回归一部分，经过多次最终得到最终位置

为什么一次回归不好：只有当Proposal和Ground Truth比较接近时（线性问题），我们才能将其作为训练样本训练我们的线性回归模型，否则会导致训练的回归模型不work（当Proposal跟GT离得较远，就是复杂的非线性问题了，此时用线性回归建模显然不合理

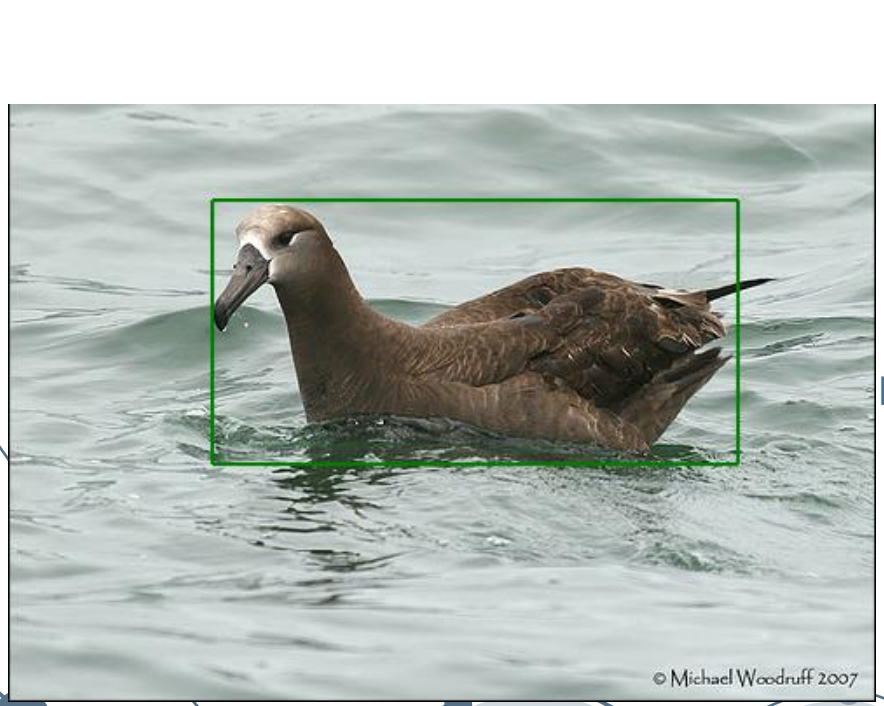
网络结构

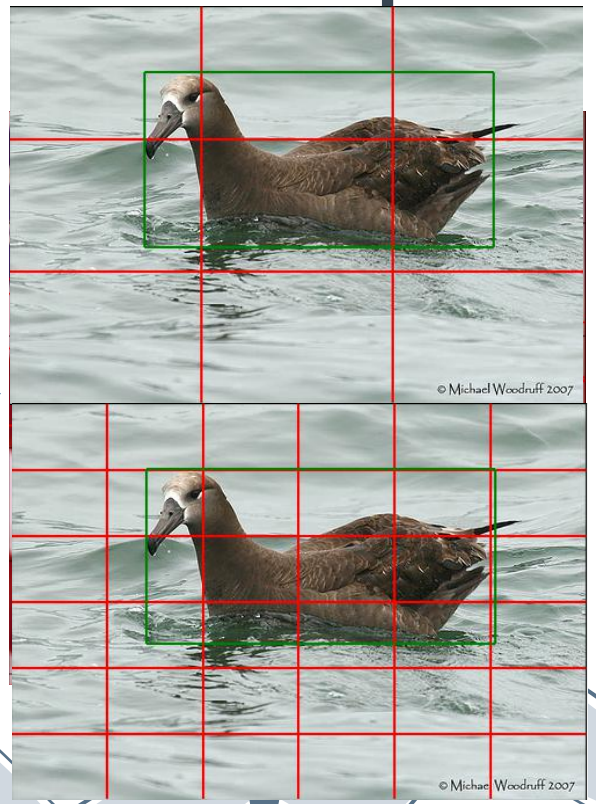




Global part is run only once to extract global features but regression part is run at every iteration.

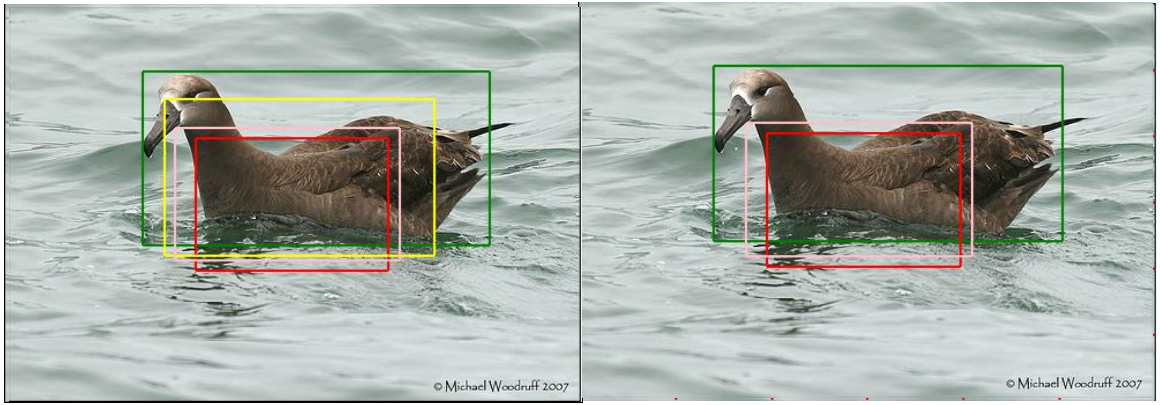
怎么代替SS的ROI Proposal





损失值计算流程

红色是原始框
粉色是回归框
黄色是阶段目标框
绿色是最终目标框



1.ROI proposal，得到 $B^1_{s=1}$ $B^2_{s=1}$ $B^3_{s=1}$
2.给$B^i_{s=1}$对应 G 。并根据iou threshold>0.2挑出部分$B^i_{s=1}$作为初始B。
3.送入网络，得到Bounding-box 回归系数t，由此我们可以得到一个新的矩形框

$$t_x = \frac{G_x - P_x}{P_w}$$
$$t_y = \frac{G_y - P_y}{P_h}$$
$$t_w = \log\left(\frac{G_w}{P_w}\right)$$
$$t_h = \log\left(\frac{G_h}{P_h}\right)$$

For s in S：
    If not s==1：
        更新B_s = T_{s-1}

4.获得当前步target框T

$$\Phi(\mathbf{B}_i^s, \mathbf{G}_i^*, s) = \mathbf{B}_i^s + \frac{\mathbf{G}_i^* - \mathbf{B}_i^s}{S_{train} - s + 1}$$

5.计算损失值

$$L_{reg}(\delta^s_{i,l_i} - \Delta(\mathbf{B}_i^s, \Phi(\mathbf{B}_i^s, \mathcal{A}(\mathbf{B}_i^s), s)))$$

$$L_{loc}(t^u, v) = \sum_{i=1}^{4} \text{smooth}_{L_1}(t_i^u - v_i)$$

6.损失值求和

模型压缩

移动设备：算不好

穿戴设备：算不了

模型压缩

1. 张量分解

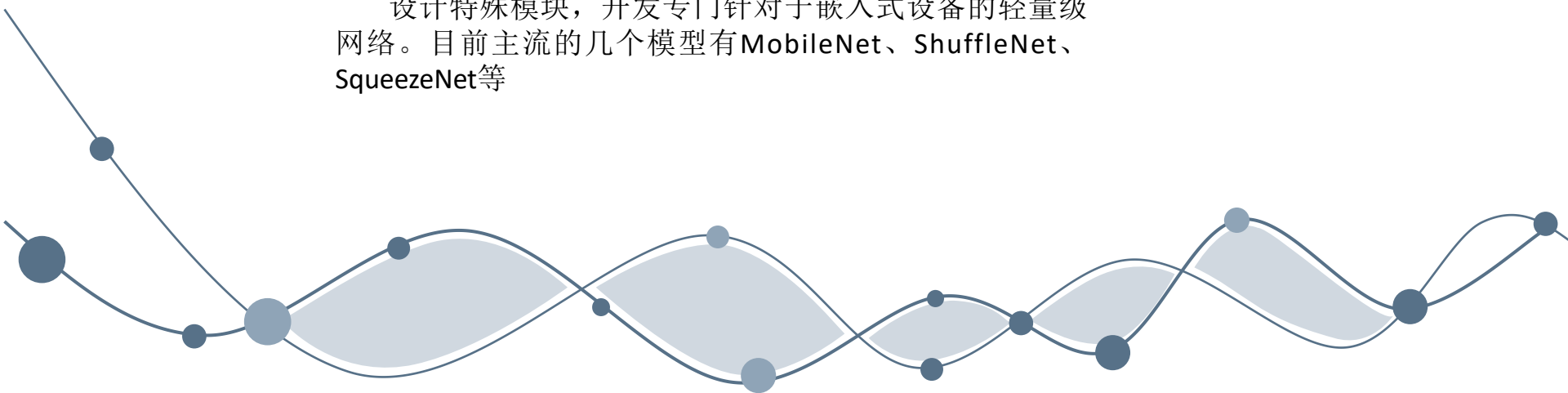
基于张量的低秩近似理论和方法，将原始的权重张量分解为两个或者多个张量，并对分解张量进行优化调整.

2. 知识蒸馏

主要利用训练完毕的复杂模型指导简单模型的训练。简单模型在训练中获取的信息量更丰富，有利于提高模型训练速度和模型性能。该方法又称教师-学生网络

3. 开发专用、轻量级模型

设计特殊模块，开发专门针对于嵌入式设备的轻量级网络。目前主流的几个模型有MobileNet、ShuffleNet、SqueezeNet等



方法综述

4. 量化

量化就是将网络的参数和中间结果用低比特来表示，简化每次运算的复杂程度，能大幅降低运算时间。目前主要有两种实现方式，一种是在训练阶段就将整个网络量化，另一种是在32bit的模型基础之上进行量化

5. 模型裁剪

模型裁剪通常对训练完毕的深度神经网络模型进行处理，其核心是寻找确定模型参数重要性的判别依据，将不重要的网络连接关系删除

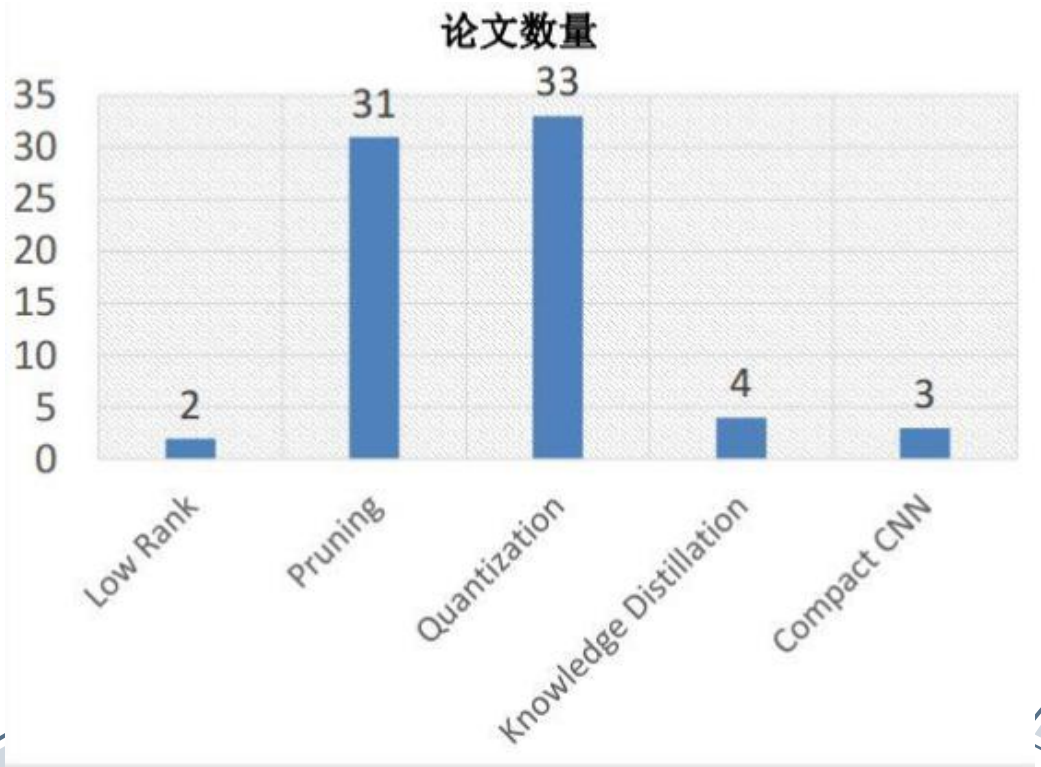
a.细粒度裁剪

通过把部分参数变成0来让模型存储体积更小。但是由于GPU进行矩阵计算的特点，稀疏化对运算速度的提升很小

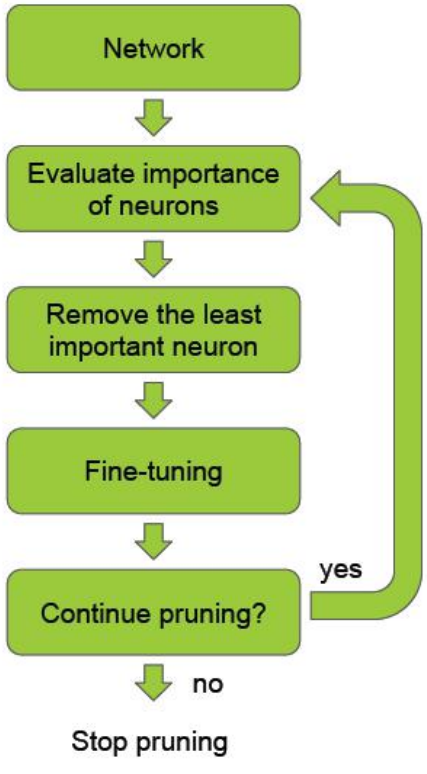
b.粗粒度裁剪

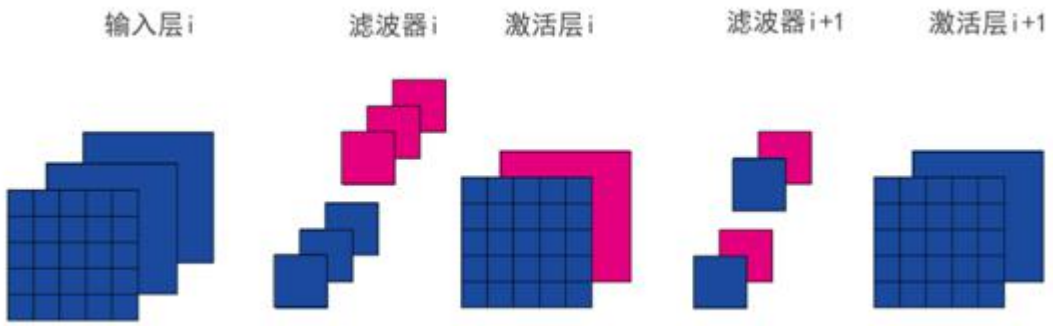
通过一种评价标准，直接裁剪掉冗余的filters，能够大幅减少计算量从而提高网络的效率。是目前比较主流的加速方法。

会议论文分布


论文数量
35
30
25
20
15
10
5
0
2
31
33
4
3
Low Rank
Pruning
Quantization
Knowledge Distillation
Compact CNN

裁剪流程


Network
Evaluate importance of neurons
Remove the least important neuron
Fine-tuning
Continue pruning?
yes
no
Stop pruning

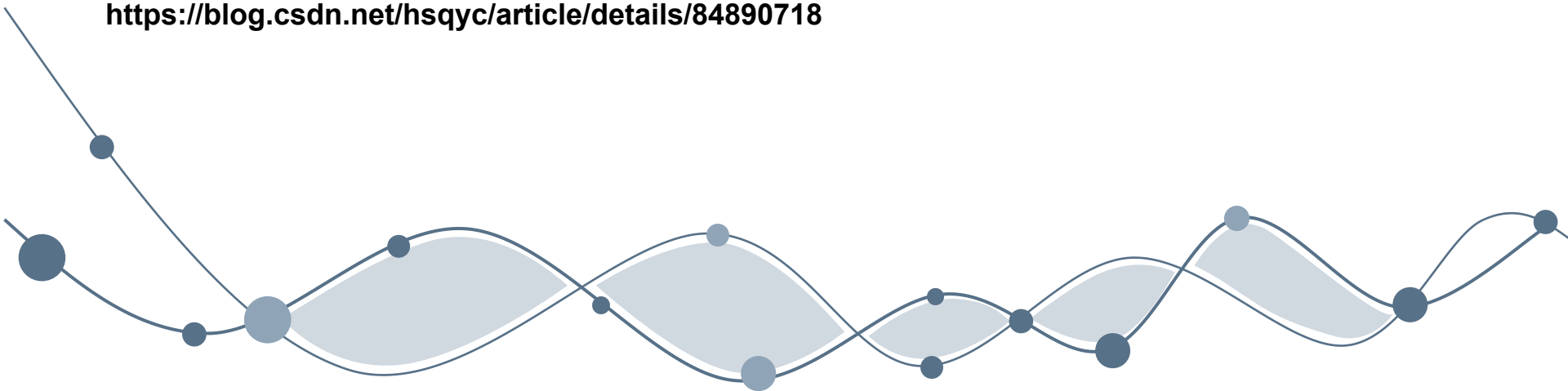

输入层i
滤波器i
激活层i
滤波器i+1
激活层i+1

裁剪论文

1.Pruning filters for effecient convnets https://blog.csdn.net/hsqyc/article/details/84029360

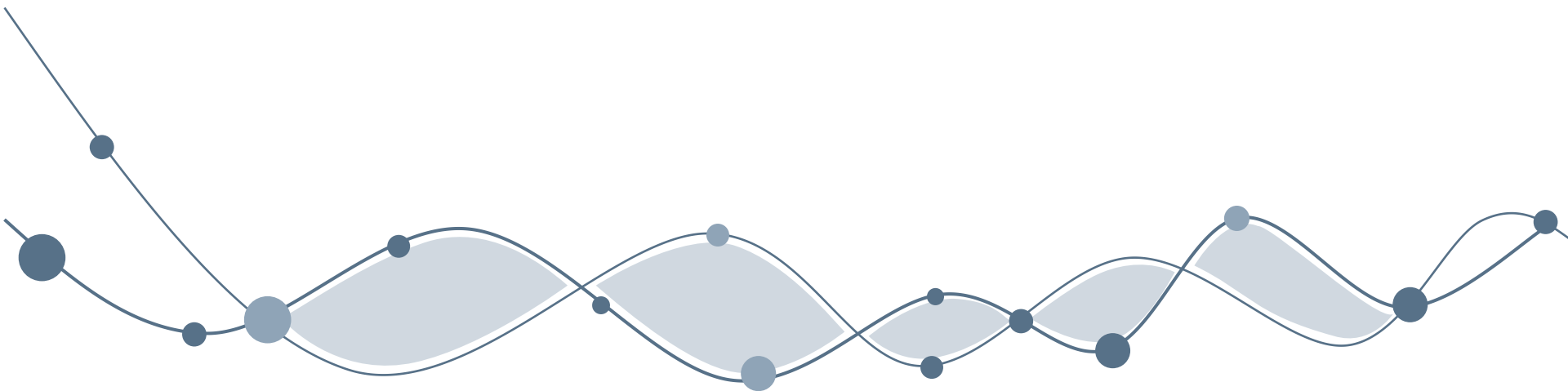
2.Network Trimming: A Data-Driven Neuron Pruning Approach towards Efficient Deep Architectures
https://blog.csdn.net/hsqyc/article/details/83651795

3.Pruning Convolutional Neural Networks for Resource Efficient Inference
https://blog.csdn.net/hsqyc/article/details/84890718



网络引用

http://www.sohu.com/a/232047203_473283
https://www.cnblogs.com/dudumiaomiao/p/6560841.html
https://www.cnblogs.com/skyfsm/p/6806246.html



演示完毕 谢谢欣赏