

# 神经激活簇: 卷积网络的无监督部件模型

## 摘要

物体类别(object categories)的部件模型对挑战识别任务来说至关重要，其中类别的差异是微妙的，并且仅反映在物体的小部件外观中。我们提出一种完全无监督方式学习部件模型的方法，无需部件注释，甚至学习期间不需要给定边界框。关键思想是使用卷积神经网络计算找到神经激活簇的模式。在我们的试验中，我们在 CUB200-2011, NA birds, Oxford PETS, and Oxford Flowers dataset 上优于现有的细粒度识别方法(无分布注释和边框标记)，并且实现了 Stanford Dog dataset 上的最好性能。我们还展示了，神经簇模型作为微调数据增强技术的优点。此外，我们的论文综合了通用和细粒度分类领域，因此我们的方法在这两种情况都适用。

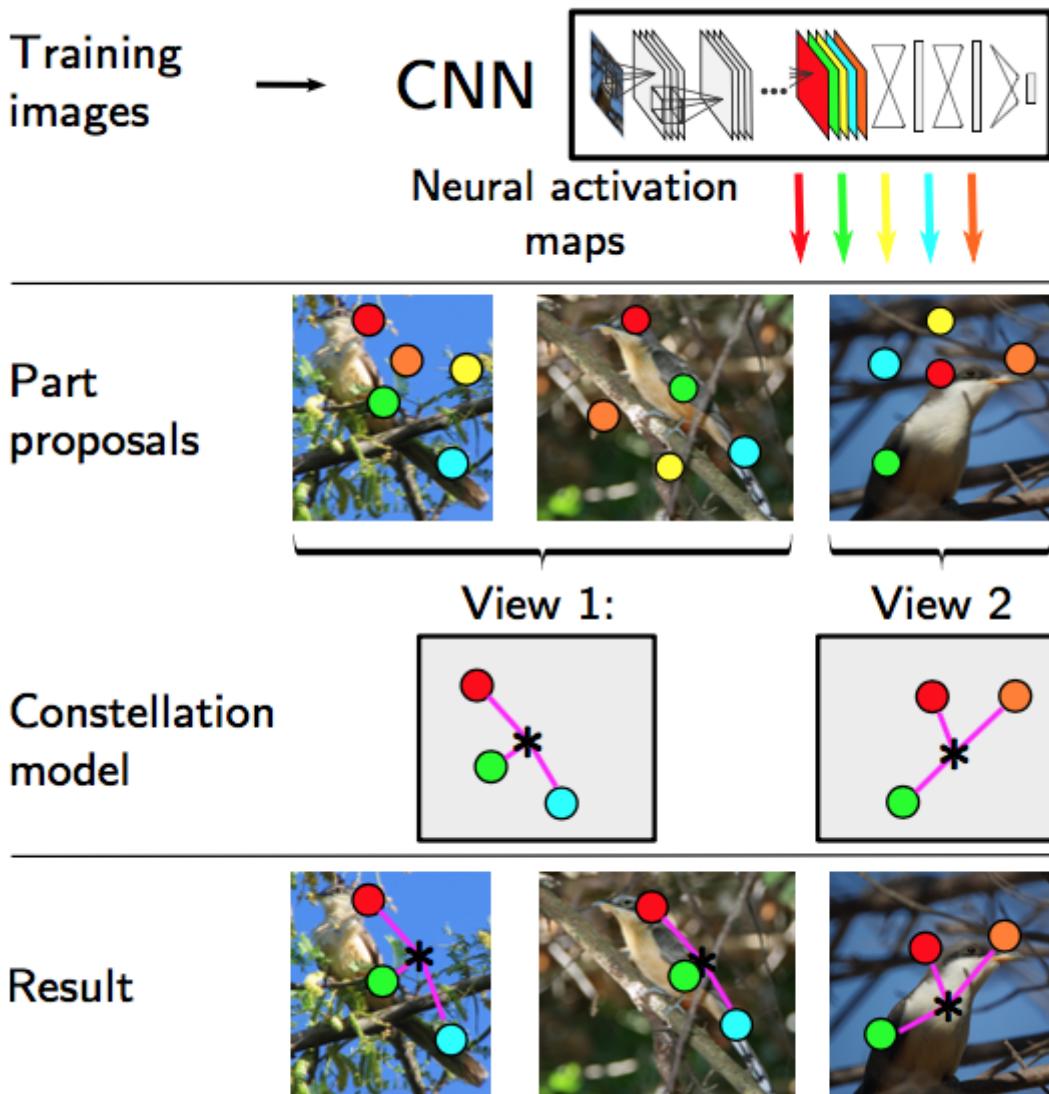


图1，我们的方法总览。深度神经激活图(Deep neural activation map)，利用CNN的通道作为部件检测器。我们通过在完全无监督数据中选择相似相对位置处的热点部件检测器来构建模型。然后使用所创建的部件模型来提取对象部件处的特征以用于弱监督分类。

## 1. 介绍

部件模型在最近众多细粒度识别方式中扮演重要角色。这种方法可以捕获物体非常局部的区别特征[18]。部件模型的学习通常通过提供部件注释或有标签的边界框，以有监督的方式完成[15, 29]。

与此相反，我们展示如何以完全无监督的方式学习部件模型，这大大降低了学习的标注成本。我们的方法是从预学习的卷积神经网络中学习神经激活簇模式。图1，显示了我们方法的总览。我们半假设我们要计算的神经激活图是CNN层的直接输出。通过随机选择部件假设的子集来构建无监督部件模型，或通过估计生成空间部件模型的参数来无监督学习部件模型。在后一种情况下，我们隐喻的发现，部件假设的子集，在图像中的某个确定簇内持续的“fire”。

虽然为部件(parts)的空间关系创建模型已经在十年前引入[16, 14]，这些方法面临巨大困难，因为部件方法(part proposals)基于人工设计的局部描述和没有对应关系的检测器。我们通过使用预学习CNN的隐部件检测器(implicit part detectors)来克服这个问题，同时大大简化了部件模型的训练。如[38]论文描述，中间CNN输出通常可以链接到公共对象的语义部件(semantic parts)，因此我们使用它们作为部件方法(part proposals)。我们的部件模型必须从已经高质量池化的部件提案(part proposals)选择每个对象视图的少数部件进行学习。这允许创建更简单和更快的部件模型，而不需要像以前工作那样明确考虑单个部件(individual parts)的外观。同时，我们不需要标注好的真实的部件定位(ground-truth part location)或边界框。

这种获得方法和学习算法在没有任何实际标记或边界注释的条件下，把细粒度识别在三个数据集上(包括CUB200-2011)提高到先进水平。此外，我们展示了如何在Caltech-256数据集上使用相同的方法进行对象识别。这是与以前的细粒度识别方法的主要区别，因为以前的大多数方法不能直接适用于其他任务。例如，我们的方法不要需要对不同尺寸的图像进行高昂的密度评估就能够在Caltech-256数据集上达到先进水平。

此外，我们的工作涉及超细粒度识别，因为我们的方法可以用来指导图像分类微调期间的数据增加。我们的实验表明，它甚至产生一个与具有实际和边框对象的细粒度CNN相比，更有识别力(more discriminative)的CNN。在下一节中，我们简要概述了部件簇模型(part constellation models)和细粒度分类领域的最新方法。

## 2. 相关工作

### 部件簇模型

部件簇模型(part constellation models)描述对象部件(object parts)之间的空间关系。部件模型学习中有很多依赖标注好的真实的部件信息(ground-truth part)或边框标记的有监督学习方式[41, 18, 29]。但是，通常标记无法获得，或者获取成本高昂。相比之下，无监督学习不需要设置任何标记，而是依赖于部件方法(part proposals)。和有监督学习的设置最大的不同在于对有用部件的选择的重要性。我们专注于无监督学习，因为这些与我们的工作紧密相关。

我们在该领域最早期的工作是[42]，其中通过用耦合的射线模型融合单个检测来进行面部角点检测(facial landmark detection)。与我们的方法类似，使用公共参考点，并且通过其相对极坐标的分布来描述其他部件的位置。但是，他们依赖于手工注释，而我们专注于无监督设置。后来，Fergus 等人[16]和Fei-Fei等人[14]基于通用SIFT兴趣点检测器构建了模型。该模型包括对象部件(object parts)的相对位置，以及相对比例和外观。虽然他们的兴趣点检测器提供了无语义(without any semantics)的多检测，但我们使用的每个基于CNN的部件检测器已经对应了特定部件对象意思(specific object part proposals)。这使我们能够更有效的设计部件选择(part selection)并加快推理。运行时复杂度与[16, 14]相比，从部件模型数目的指数(exponential in the number of modeled parts)，减少到线性时间复杂度。类似的计算局限(computational limitations)也出现在其他作品中。特别是在大量部件方法(part proposals)下，具有明显的优点。

Yang 等人[37]从一组随机初始化的图像补丁(image patches)中选择对象部件模板。他们基于一组训练图像中的模板的同现(co-occurrence)，多样性，适合度来构建部件模型。检测到的对象部件，用于鸟的基于部件的细粒度分类。在我们的应用中，同现和适应性是基于CNN部件方式(part proposals)中相当弱的属性。例如，频繁出现的背景图案(如树的叶子)，检测器可以通过其算法来选择。相反，我们的工作考虑空间关系，用来过滤在不一致的相对位置上出发的不相关的背景检测器。

Crandall 等人[11]通过联合考虑对象和场景相关部件来改进部件模型学习。然而，对象和不同背景图案的组合成的视图数目是巨大的。相比之下，我们基于相对位置选择的方法更简单和有效，因为我们只想识别用于分类的有用部件方法(part proposals)。

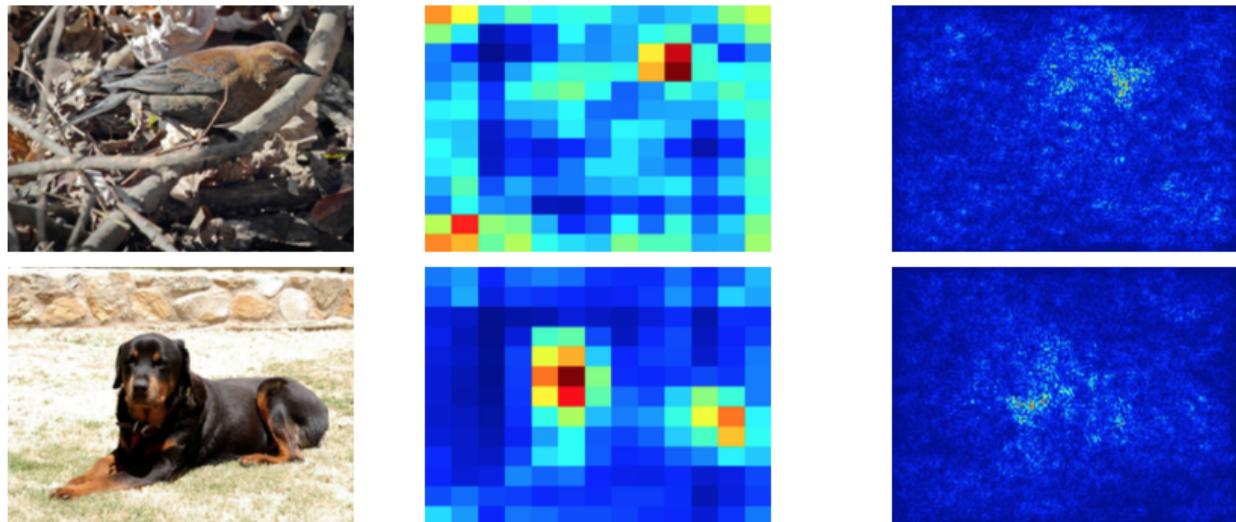
在检测领域中，有很多基于部件模型的方法。deformable part model(DPM)是其中最流行的一个。他使用潜在鉴别(latent discriminative)SVM模型学习具有边框的部件簇模型。大多数检测方法至少需要真是边框标注。相比之下，我们的方法不需要这样的注释，或者任何负样本，因为我们使用生成模型学习簇模型(constellation model)并且使用无边框的对象部件方法(object part proposals)。

## 基于部件模型的细粒度识别

精细识别重点研究在视觉上非常相似的类，其中不同的对象类别有时仅在细微的细节上不同。例如，鸟类的识别[35]或车模型识别。由于对象的小部件差异很重要，使用部件模型的局部特征提取具有重要作用。

在细粒度识别领域中的早期工作之一就是使用椭圆体来建模鸟的姿势，并使用特定核函数来融合获得的部件。其他工程建立在DPM上。例如，DPDM(deformable part descriptor method)[41]使用有监督版本的[15]去训练DPM(deformable part models)，其允许通过比较对应的部件进行姿态归一化。[17]和[18]的工作证明了细粒度识别的非参数部件检测。基本思想是从最近邻匹配得的相似训练样本中获取转换为人工标注部件位置。Chai 等人[8] 使用DPM的检测和GrabCut的分割输出预测不见位置。Branson 等人[7] 使用部件位置(part location)弯曲图像块(image patches)成姿态归一化表示。Zhang 等人[39] 从选择性搜索[33]生成的对象方式(object proposals)中选择对象部件检测。所提及的方法使用所获得的部件位置来计算局部特征。Berg 等人[4]为每对部件和类学习一个线性分类器。众多分类器的判定值被用作特征表示。虽然所有这些方法在许多任务中运行良好，但他们通常在训练和测试时需要真实值标注。相比之下，我们的方法不需要昂贵的注释部件位置，而是在完全无监督下进行部件模型学习。这也紧随着最近逐渐减少训练期间注释的潮流[39, 36, 29]。Simon 等人[29]的方法，在训练期间需要对象的边框而不是部件标注。他们还利用神经激活图进行部件发现(part discovery)，虽然我们的方法不需要边界框，当我们仍能够改进他们的结果。

我们处理的无监督情景，Xiao 等人[36]也有考虑。他们聚集CNN的最后一个卷积层的通道成组。基于每个组的激活来提取对象和每个部件的斑块信息。该补丁用来分类图像。然而他们的工作需要为感兴趣的对象预训练分类，而我们只需要一个可以在弱相关对象数据集上进行预训练的CNN。



Input	CNN last conv. output	Neural activation map
$I$	$\begin{bmatrix} f_{1,1} & \dots & f_{1,13} \\ \dots & \dots & \dots \\ f_{13,1} & \dots & f_{13,13} \end{bmatrix}$	$\begin{bmatrix} m_{1,1} & \dots & m_{1,227} \\ \dots & \dots & \dots \\ m_{227,1} & \dots & m_{227,227} \end{bmatrix}$

最后卷积层的通道输出示例，和两个图像对应的神经激活图(跳过通道的索引一简化符号)。深红色表示高度

活跃，深蓝色表示完全不活跃。激活图具有更高的分辨率，更适合于部件定位。Best viewed in color.

### 3. 深度神经激活图

CNN已经展示了从零开始学习一个完整分类管道的巨大潜力，而不需要手动定义低级特征。最近的CNN架构[22, 31]包括多层卷积，池操作，完全线性变换和非线性激活。

卷积层将输入与多个内核卷积。如[38]所示，前层(early layers)中的卷积内核类似于许多流行的低级特征描述符(如HOG或SIFT)中使用的滤波掩膜。他们的工作还表明，越后面的层对越抽象的图像敏感。这些模式，甚至可以对应于整个对象[30]或对象的部件[29]，这正是我们要利用的。

全连接层之前的一层输出  $f$  以二维输出元素的相识排列在多个通道  $1 \leq p \leq P$  中，例如，我们通过  $(f_{j,j'}^p(I))$  表示  $f$  其中  $I \in \mathbb{R}^{W \times H}$  表示输入图像，而  $j$  和  $j'$  表示通道中的输出元素。图2，展示了最后卷积层通道输出的例子。可以看出，输出可以理解为多个对象部件检测器的检测分数。因此，CNN自动学习隐式部件检测器与其训练的数据集相关。在这种情况下，可视化通道在鸟头和狗头部件显示了高输出。

通道输出的缺点是其分辨率，无法做到部件的精确定位。因此，我们遵循[30]和[29]的基本思想，并计算了深度神经激活图。我们计算了输出通道  $p$ ，平均输出的梯度，其中  $I_{xy}$  表示输入图像像素。

$$m_{x,y}^p(I) = \frac{\partial}{\partial I_{x,y}} \sum_{j,j'} f_{j,j'}^{(p)}(I)$$

计算输出很容易通过BP (backpropagation pass)实现。梯度的绝对值显示图像中的哪些像素对通道的输出有最大的影响。与层的实际输出类似，他能够定位出该通道的敏感图像。然而，深度神经激活图的分辨率要更高一些(图2)。在我们的试验中，我们通过使用激活图上的最大激活点来计算训练图像  $I_i$  的有意义部件位置(part proposals location)。

$$\mu_{i,p} = \operatorname{argmax}_{xy} |m_{x,y}^{(p)}(I_i)|$$

CNN的每个通道为每个图像提供一个神经激活图，因此我们可以从每个通道获得一个有意义部件提案(part proposals)。通过添加每个输入通道的绝对激活图来处理RGB图像。因此，我们将深度神经激活图减少为2D坐标，并且在部件模型学习期间不考虑各个部件的图像块。然而，在分类时，在预测部件位置处提取的图像块被用来做特征检测。

隐式部件检测器在CNN训练期间自动学习。这与其他部件发现方法(例如poselets[6])相比是一个巨大的优势，那些方法无法不一定产生对先验分类有用的部件。在我们的例子中，用于训练CNN的数据集并不一定需要与最终数据集相同，以及我们想要构建的部件表示任务相同。

此外，计算部件提案(part proposals)几乎和使用CNN一样快(在带有GPU的标准电脑上对每图10个部件仅要110ms)，这能够做到实时应用。我们的项目主页上提供了一个基于此，以10fps运行的鸟头探测器的视频。在下面的论文中我们继续介绍部件组件(part proposals)。

## 4. 无监督部件模型检测

在这一节，我们展示了给出一组目标类的训练图像集，如何用无监督的方式有效地构建部件模型。结果部件模型被用作特征提取定位和随后的细粒度分类。和许多之前的工作相比，我们有一套鲁棒的但不是必须的相关的部件提案和为当前的目标类选择有用的那个部件。其它方法。主要的结果是我们不需要关心稳定的部件探测器的昂贵的训练。我们的任务使有用探测器的筛选变得精简。

作为输入，我们对训练图像  $i = 1, \dots, N$  和部件提案  $p = 1, \dots, P$  和使用归一化的部件提案位置  $\mu_{i,p} \in [0, 1]^2$ 。P部件提案与在一个CNN中间的输出层通道相符合。 $\mu_{i,p}$ 通过计算通道p的激活地图，如果一个通道的激活地图平均趋向于0，部件提议就会被考虑隐藏起来。由于整流线性单元被用作非线性激活所以这种稀疏性自然地就发生了。

### 4.1 部件的随机选择

构造一个部件模型的一种简单方法是从全部的P个提案中选择随机M个部件。对于所有的训练图片，我们挑选出描述斑块位置周围的图像区域的M维特征向量，特征堆叠和一个线性SVM被学习用作图像标签。这些甚至可以结合微调CNN被用来提取斑块特征。关于部件特征的更多细节将在第5节给出。

在我们的试验中，我们展示的对于普通对象识别随机识别确实是一种有效的技术。然而，对于细粒度识别，我们需要选择可能对应于同一种对象但不是一个背景的部件。使用所有的提案不是一个选择，因为特征描述明显地增加了使训练不切实际。因此，我们在接下来展示如何选择几个部件来提高分类性能并减少特征计算的计算时间。

### 4.2 Constellations of neural activations

目标是使用所有训练图像的所有部件提案的2D位置来对被选择的提案的一个子集估算一个星型模型。同其它受欢迎的部件模型例如DPM一样，我们的模型也合并了感兴趣对象的多种视图  $v = 1, \dots, V$ 。例如，一辆车的前视图和侧视图是不一样的，那么每一种视图就需要不同的部件来描述。

每一个视图都包括由指示变量  $b_{v,p} \in [0, 1]$  定义的M个部件提案，我们称它们为部件(parts)。此外，还有一套相对应的位移矢量  $d_{v,p} \in [-1, 1]^2$ 。位移矢量是从部件p到一幅图像i中的对象的共同位置  $a_i$  的理想相对偏移量。因为在学习过程中没有对象注释给出，所以  $a_i$  是一个隐变量 (latent variables)。

另外一组隐变量  $s_{i,v} \in [0, 1]$  表示每张训练图像的视角选择。我们假设每张图像只有一个目标对象可见，因此对于每幅图像仅选择一种视图。最后， $h_{i,p} \in [0, 1]$  表示如果部件p在图像i中可见。在我们的实验中，一个部件的可见性由部件提案决定而不是在学习中估计。

**学习目标 (Learning objective)** 我们通过所有模型的最大后验估计和从部件提案位置  $\mu$  提供的隐参数  $\tau = (b, d, s, a)$  来确定训练图像的最佳模型：

$$\tau = \operatorname{argmax}_\tau p(\tau | \mu)$$

与隐变量边缘化不同的是，我们获得一个非常有效的学习算法。我们应用贝叶斯准则，假设训练图片和部件提案给出的模型参数是不相关的。假设  $a$  和  $d$  是平等先验， $b$  和  $s$  是不相关先验。

$$\operatorname{argmax}_\tau p(\mu | b, d, s, a) \cdot p(b) \cdot p(s) = \operatorname{argmax}_\tau \prod_{i=1}^N \left( \prod_{p=1}^P p(\mu_{i,p} | b, d, s, a) \right) \cdot p(b) \cdot p(s)$$

$p(\mu | b, d, s, a)$  是给定模型的预测部件位置定位的分布。如果部件  $p$  在图像  $i$  的视角  $v$  中被使用，我们假设部件定位是在 root location 加上移位矢量周围正常地分布，

$\mu_{i,p} \sim N(d_{v,p} + a_i, \sigma_{v,p}^2 E)$  表示单位矩阵。如果部件没有被使用，关于定位的位置没有任何先验信息，我们假设它均匀分布在  $I_i$  中的多有可能的图片位置。

分布公式如下

$$p(\mu_{i,p} | b, d, s, a) = \prod_{v=1}^V N(\mu_{i,p} | a_i + d_{v,p}, \sigma_{v,p}^2 E)^{t_{i,v,p}} \left( \frac{1}{|I_i|} \right)^{1-t_{i,v,p}}$$

$t_{i,v,p} = s_{i,v} b_{v,p} h_{i,p} \epsilon [0, 1]$  表示无论部件  $p$  在视图  $v$  中被使用或者是可见，它在图像  $i$  中本身就是激活的。部件选择  $b$  先验分布只能捕获  $M$  个部件需要被选择的约束，即

$\forall v : M = \sum_{p=1}^P b_{v,p}$ . 视角选择  $s$  的先验包含在训练图片  $i$  中只有一个单独的视图是激活的假设，即  $\forall i : 1 = \sum_{v=1}^V s_{i,v}$

一般来说，我们表示可行的变量集合为  $M$ 。利用这些和应用 log 进一步简化公式 4：

$$\operatorname{argmin}_{\tau \in M} - \sum_{i=1}^N \sum_{p=1}^P \sum_{v=1}^V t_{i,v,p} \log N(\mu_{i,p} | a_i + d_{v,p}, \sigma_{v,p}^2)$$

此外，我们假设方差对所有视图的所有部件是恒定不变的。因此，最终优化问题的表达式是

$$\operatorname{argmin}_{\tau \in M} - \sum_{i=1}^N \sum_{p=1}^P \sum_{v=1}^V s_{i,v} b_{v,p} h_{i,p} \|\mu_{i,p} - a_i - d_{v,p}\|^2$$

**优化 (optimization)** 公式 6 独立地交替优化每个模型的变量  $b$  和  $d$ ，以及隐变量  $a$  和  $s$ ，类似于 EM 算法。对于每一组  $b$  和  $s$  变量，我们可以通过误差项排序来计算最值。例如， $b_{v,p}$  由分析可以计算

$$\operatorname{argmin}_{b \in \tau_b} \sum_{p=1}^P \sum_{v=1}^V b_{v,p} \left( \sum_{i=1}^N s_{i,v} h_{i,p} \| \mu_{i,p} - a_i - d_{v,p} \|^2 \right)$$

这种优化可以直观地解决。首先，每一种视角都是独立考虑的，对每一个视角我们选择固定数量的部件而不考虑其它的东西。对每一种部件提案，我们计算它的 $E(v, p)$ 。这一项描述了部件提案 $p$ 和视角 $v$ 的适合程度。如果它的值很小，那么部件提案与视角相适合这个提案应该被选择。我们现在计算视角 $v$ 的所有部件的 $E(v, p)$ ，选择最小的 $M$ 个部件。以同样的方法，视角选择 $s$ 也可以决定出来。

固定 $b, s$ 和 $d$ 获得根节点 $a$ :

$$\hat{a}_i = \sum_{v,p} (\mu_i^p - d_{v,p}) / (\sum_{v,p} t_{i,v,p})$$

同样地，我们获得位移向量 $d_{v,p}$ :

$$d_{v,p} = \sum_{i=1}^N t_{i',v,p} \cdot (\mu_{i,p} - a_i) / (\sum_{i'=1}^N t_{i',v,p})$$

公式是直观的，例如位移向量 $d_{v,p}$ 由根节点 $a_i$ 和预测部件定位位置 $\mu_{i,p}$ 之间的平均偏移量确定。然而平均值只使用图像中被用的部件 $p$ 来计算。

这种优化方法与EM算法相比具有相同的挑战性。特别是变量的初始化是至关重要的。我们初始化 $a$ 为图像的中心， $s$ 和 $b$ 表示对视角和对每一种视角所选择的部件的随机分配。 $d$ 的初始化通过计算它的第一个值获得。 $b$ 的值被用来确定收敛。使用不同的初始化重复此优化并使用具有最佳目标值的结果。

推理（inference）对于一个未知的测试图像的推理步骤与在训练过程中的计算方法相似。通过固定学到的模型参数 $b$ 和 $d$ 通过公式7和8迭代估计参数 $s$ 和 $a$ 。通过神经激活地图可见性可以再一次直接看见。

## 5. 实验

实验包括三个主要的方面和我们的方法应用。首先，我们提出了一个基于我们的微调方法的部件模型的数据增强技术，优于在边界框上微调。第二，我们将我们的方法应用到几个细粒度分类，大多数现有的方法依赖于部件注释标注数据。最后，我们展示了如何使用相同的方法为普通的图像分类，也表明了这方面的优势。我们方法的代码是公开可获得的。

### 5.1 实验步骤

**数据集** 我们在实验中使用五个不同的数据集。对细粒度分类，我们在CUB200-2011 (200 classes, 11788 images)上评估了我们的方法，NA birds (555 classes, 48562 images), Stanford dogs (120 classes, 20580 images), Oxford flowers 102 (102 classes, 8189 images), and Oxford-IIIT Pets (37 classes, 7349 images)。我们使用提供好的已划分成训练

和测试的数据集，遵循相应论文的评估协议。我们展示了在CUB200-2001上的正确率和在其它数据集上平均分类的正确率。对于普通对象识别任务，我们在Caltech256上做了评估，这个数据集包含256类对象30607张图片。我们遵循[31]的评估协议。随机选择了60训练图像其它的用做测试。

CNN和参数 在我们的试验中使用了两种不同的CNN架构：广泛使用的是Krizhevsky的AlexNet架构，更准确的是Simonyan的。在NA鸟的事例中我们使用了GoogLeNet网络。关于网络结构更多的细节，我们请读者阅读相应的论文。值得注意的是我们的方法可以用在任何CNN网络中。使用relu6, relu7和pool5/7x7 s1层分别进行特征计算。部件位置的定位使用的是pool5层。这些层由256和512个通道组成，分别产生256和512个部件提案。在CUB200-2011试验中，NA鸟、牛津狗，宠物和花卉的数据集，使用的是微调我们提出的数据增强技术。我们使用两部微调方法，开始的学习率是0.001，当损失没有任何变化时下降到0.0001。在斯坦福狗的试验中，它是ILSVRC 2012训练图像集的一个子集，CNNs预训练网络的评估ILSVRC 2012 图像上被偏置为完整的数据集。因此我们删除了ILSVRC 2012 训练集下斯坦福狗的测试图像，从零开始在这些修改后的数据集中学习到一个CNN。训练模型从我们的网页中可以获取，你可以轻松比较这项工作。

如果没有另外提及，学习到的部件模型使用了5个视角，每个视角有10个部件。对于每一类分别学习到一个模型。部件模型学习重复五次，我们就可以得到最佳目标值的模型。我们计算每个部件使用多少图像，在分类时选择10个最经常选择到的部件来使用。

分类框架 我们使用基于部件分类方法，这种方法是Simon提出的。给定所有被选择部件的预测定位后，我们以每个部件为中心裁剪正方形边框，计算他们多有的特征。这些边框的尺寸由 $\sqrt{\lambda \cdot W \cdot H}$ ,  $\lambda \epsilon (\frac{1}{5}, \frac{1}{16})$ .

给定，W和H分别是未裁剪图像的宽和高。如果部件不可见，使用一张平均图像上的特征计算作为替代。这种填补方法与零填补具有相似的性能，但是在某些情况下会产生轻微的性能增益。在CUB200-2011试验中，我们也估算了每一张图像的边界框。选择性搜索应用在每一张图像上来产生边界提案。每一个提案由CNN和具有最高分类置信度的提案决定，被用作估计边界框。

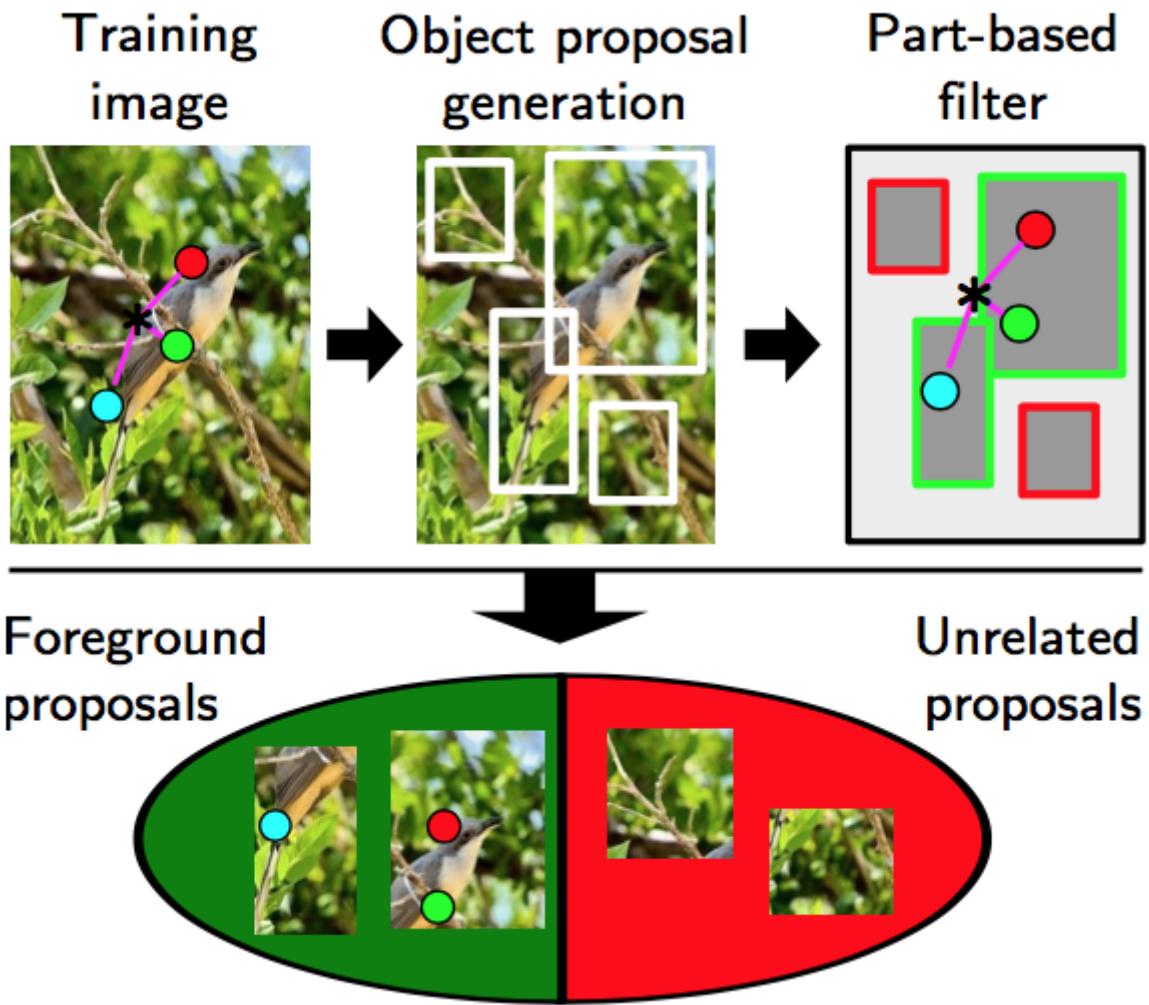
每一部件的特征、未裁剪图像和已估计边界框是重叠的，并且使用线性SVM来分类。在CUB200-2011试验中，也使用翻转的训练图像。超参数在CUB200-2011训练数据上通过交叉验证优化，也可以用其它数据集。

## 5.2 使用部件提案增加数据

微调是一个预学习CNN网络到一个特定领域数据集的适应过程。它显著地提高了许多工作的性能。因为特定领域的数据集通常很小，因此CNN的训练很容易发生过拟合，通过“数据增加”人为地扩大训练集，例如[22,31]是一个常用的技术，随机地剪裁一个固定的大尺寸的图像斑块。如果训练图像被剪切到感兴趣的对象，这种方法特别有效。如果图像没有被建材，没有可获得的注释边界框，未剪裁的图像可以作为代替被使用。然而，像表1展示的一样微调效果较差。因为注释边界框经常不能获取或者获取的代价非常昂贵，我们建议对目标提案过滤

器微调用一个新颖地选择体系来代替。

我们的方法概述如图3所示。



首先，我们为每一张训练图像选择相对应视图的最适合模型的五个部件。第二，许多对象提案的生成使用选择搜索。这些天使非常嘈杂的。许多只包含背景而没有感兴趣对象。我们计算有多少预测部件包含在每个提案中，只选择至少包含三个部件提案。剩余的斑块，在平均情况下约为48的CUB200-2011，是包含感兴趣对象的高质量图像区域。最后，我们使用微调，过滤所有训练图像的提案。

这些方法的结果在表1中展示。微调这些斑块不仅与在剪裁图像上相比是一个增益，它也消除了边界注释框的需要。

Train. Anno.	Method	Accuracy
Bbox	Fine-tuning on cropped images	67.24%
None	No fine-tuning	63.77%
None	Fine-tuning on uncropped images	66.10%
<b>None</b>	Fine-tuning on filtered part proposals	<b>67.97%</b>

**Table 1.** Influence of the augmentation technique used for fine-tuning in case of AlexNet on CUB200-2011. Classification accuracies were obtained by using 8 parts as described in Sect. 5.3.

### 5.3 没有注释的细粒度分类

在细粒度分类领域的大多数方法依赖像注释部件位置或者边界框这样的额外的注释。最近的工作将几个根据需求的注释量的多少区分开来。这些方法也使用了部件注释，只标注边界框或者一点注释也没有。另外，训练中需要的注释和测试时需要的注释区分开。我们的方法仅使用训练图像的类标签，没有额外的注释。

CUB200-2011 在CUB200-2011上进行细粒度识别的结果如表2所示。对每一个CNN架构我们提出了三个不同的结果。“无部件”相当于只有全局图像特征。“Ours, rand”和“Ours,const”是4.1节和4.2节提出的方法。正如表中看到的，我们的方法提高了Xiao等人的工作的3.1%，同比较少错误超过16%以上。值得一提的是他们的工作需要一个鸟类的预处理网络分类器为了选择有用的斑块进行微调。另外作者证实他们使用ImageNet中的大量的鸟类子集来预处理他们的CNN网络。相反，我们的工作更容易适应其它数据集，因为我们只需要一个通用的预训练CNN并且没有特定域之外的训练数据。我们的方法和由Simon设置的第三种最好的结果之间的差异是我们方法的准确度比他们高27%多。表格中也展示了用没有部件和随机选择部件的方法分类的正确率。正如我们看见的，随机部件选择与使用没有部件的方法相比平均提高了8%的正确率。我们提出的部件选择方案进一步提高了性能，使用AlexNet的准确率是68.5%，使用VGG19的准确率是81.01%。

Train. Anno.	Test Anno.	Method	Accuracy
Parts	Bbox	Bbox CNN features	56.00%
Parts	Bbox	Berg <i>et al.</i> [4]	56.78%
Parts	Bbox	Goering <i>et al.</i> [18]	57.84%
Parts	Bbox	Chai <i>et al.</i> [8]	59.40%
Parts	Bbox	Simon <i>et al.</i> [29]	62.53%
Parts	Bbox	Donahue <i>et al.</i> [12]	64.96%
Parts	None	Simon <i>et al.</i> [29]	60.55%
Parts	None	Zhang <i>et al.</i> [39]	73.50%
Parts	None	Branson <i>et al.</i> [7]	75.70%
Bbox	None	Simon <i>et al.</i> [29]	53.75%
None	None	Xiao <i>et al.</i> [36] (AlexNet)	69.70%
None	None	Xiao <i>et al.</i> [36] (VGG19)	77.90%
None	None	No parts (AlexNet)	52.20%
None	None	Ours, rand., Sect. 4.1 (AlexNet)	60.30 $\pm$ 0.74%
None	None	Ours, const., Sect. 4.2 (AlexNet)	68.50%
None	None	No parts (VGG19)	71.94%
None	None	Ours, rand., Sect. 4.1 (VGG19)	79.44 $\pm$ 0.56%
None	None	Ours, const., Sect. 4.2 (VGG19)	<b>81.01%</b>

Table 2. Species categorization performance on CUB200-2011.

NA birds 表3展示了我们的方法在相对新的NA鸟数据集上的结果。没有使用任何部件的正确率仅有63.9%。和CUB200-2011数据集相同的是，使用我们的部件选择方法有一个明显的优势是我们的正确率为76.3%。有趣的是，这个正确率非常接近CUB200，同时在NA birds上超过2.5倍以上。尽管我们不使用任何种类的部件注释，我们的方法也比[7]的方法要做的更好。  
Stanford dogs 表4是Stanford dogs的准确度。据我们所知，仅有一项工作展示了CNN训练的结果，不包括Stanford dogs的测试图像。Sermanent微调了他们的深度GoogleLeNet的结构得到了75%的正确率。在我们的实验中，我们使用Krizhevsky的弱得多的架构正确率仍然达到68.61%。与其他非深度网络结构相比，这意味着提高了16%多的正确率。

Train. Anno.	Test Anno.	Method	Accuracy
Parts	Parts	Horn <i>et al.</i> [34]	75.0%
None	None	No parts (GoogLeNet)	63.9%
None	None	Ours, const., Sect. 4.2 (GoogLeNet)	<b>76.3%</b>

Table 3. Species categorization performance on NA Birds.

Method	Accuracy
Chai <i>et al.</i> [8]	45.60%
Gavves <i>et al.</i> [17]	50.10%
Chen <i>et al.</i> [10]	52.00%
Google LeNet ft [28]	75.00%
No parts (AlexNet)	55.90%
Ours, rand., Sect. 4.1 (AlexNet)	$63.29 \pm 0.97\%$
Ours, const., Sect. 4.2 (AlexNet)	68.61%

Table 4. Species categorization performance on Stanford dogs.

#### Oxford pets and flowers

表5和表6是Oxford pets and flowers的结果。在两个数据集上我们的方法仍然按显著地优于之前的方法。和其它数据集类似，随机选择部件提高了4%的正确率。我们方法的意义在于显著改善了分类的正确率，并且在两个数据集上分类的正确率分别达到了95.35%和91.60%。

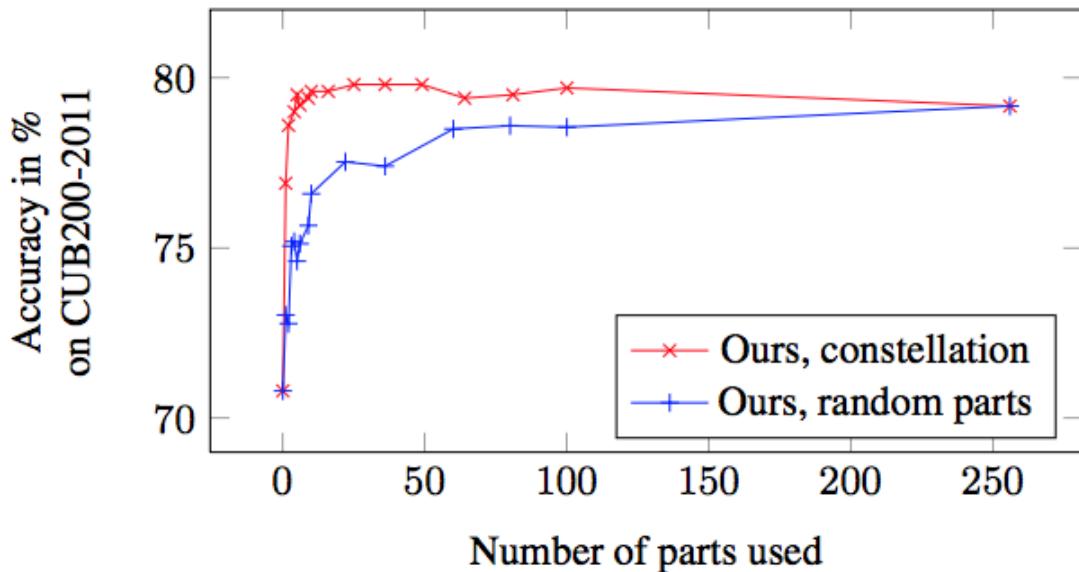
Method	Accuracy
Angelova <i>et al.</i> [2]	80.66%
Murray <i>et al.</i> [23]	84.60%
Razavian <i>et al.</i> [26]	86.80%
Azizpour <i>et al.</i> [3]	91.30%
No parts (AlexNet)	90.35%
Ours, rand., Sect. 4.1 (AlexNet)	90.32 $\pm$ 0.18%
Ours, const., Sect. 4.2 (AlexNet)	91.74%
No parts (VGG19)	93.07%
Ours, rand., Sect. 4.1 (VGG19)	94.20 $\pm$ 0.23%
Ours, const., Sect. 4.2 (VGG19)	<b>95.34%</b>

Table 5. Classification performance on Oxford 102 flowers.

Method	Accuracy
Bo <i>et al.</i> [5].	53.40%
Angelova <i>et al.</i> [2].	54.30%
Murray <i>et al.</i> [23].	56.80%
Azizpour <i>et al.</i> [3].	88.10%
No parts (AlexNet)	78.55%
Ours, rand., Sect. 4.1 (AlexNet)	82.70 $\pm$ 1.64%
Ours, const., Sect. 4.2 (AlexNet)	85.20%
No parts (VGG19)	88.76%
Ours, rand., Sect. 4.1 (VGG19)	90.42 $\pm$ 0.94%
Ours, const., Sect. 4.2 (VGG19)	<b>91.60%</b>

Table 6. Species categorization performance on Oxford-IIIT Pets.

部件数量的影响 图7提供了深入了解在分类中使用的部件数量的影响。我们比较了随机部件和基于选择性的part constellation模型。与之前的实验相比，每个部件提取一个斑块时  $\lambda = \frac{1}{10}$ 。对任意数量的部件当随机部件提高准确度时，所选择的方案显然选择更多相关的部件来帮助提高正确率。



**Table 7.** Influence of the number of parts on the accuracy on CUB200-2011. One patch was extracted for each part proposal.

## 5.4 从细粒度分类到普通分类

几乎所有目前的细粒度识别方法是特定的算法，它很难去适应现在的普通分类。主要原因是在细粒度识别中所有对象共享语义部件。这是否意味在细粒度识别领域的丰富的知识不能被其他领域所用了呢？细粒度分类和普通分类如此不同？在我们看来，答案很明显不是，我们提出的方法就是一个很好的例子。

应用细粒度分类方法到其它的任务中有两个主要的挑战。第一，语义部件探测器需要通过更抽象的兴趣点探测器所代替。第二，有用的兴趣点探测器的选择和训练需要考虑每个对象类独有的形状和语义集部件。我们的方法可以正常的应用在普通分类任务。第一个挑战已经解决，我们使用一个训练好的CNN的部件探测器来区分数量巨大的种类。因为这些特性，部件提案可以被看作带有一个特有的模型的普通兴趣点探测器。与语义部件不同的是，它们不一定只识别一个特定对象的一个特定部件。相反，它们能捕捉到许多不同种类对象的兴趣点。第二个挑战已经解决，我们创建class-wise部件模型，选择在大多数类中共享的部件提案。然而，对部件探测器的随机选择已经增加了分类的准确度。

### Caltech 256

表8展示了在Caltech 256 数据集上采用我们的方法的进行分类的结果。我们所提出的方法全局特征的基线标准，没有多重采样，在AlexNet实验中提高了1%，在VGG19试验中提高了1.6%。虽然Simonyan实现了略微高一点的性能，但是他们的方法也是更昂贵的，因为他们用所有可能的三种不同的规模裁剪图片在整个CNN上密集诊断。他们使用两种CNN模型的融合实现的最好的结果是86.2%，这与我们相应的实验没有可比性。结果清楚的显示了通过更多的普通探测器替换语义部件探测器完全可以应用在在其它领域上的细粒度分类方法。关于普通图像分类目前许多方法依赖于“Blind”部件。例如，空间金字塔或者其它采样方法是相当

于部件探测器，部件探测器总是能在一幅图像的固定位置探测到一些东西。结合更加复杂的多类部件模型替换这些“blind”探测器是一个自然地改善。

Method	Accuracy
Zeiler <i>et al.</i> [38]	74.20%
Chatfield <i>et al.</i> [9]	78.82%
Simonyan <i>et al.</i> [31] + VGG19	85.10%
No parts (AlexNet)	71.44%
Ours, rand., Sect. 4.1 (AlexNet)	72.39%
Ours, const., Sect. 4.2 (AlexNet)	72.57%
No parts (VGG19)	82.44%
Ours, const., Sect. 4.2 (VGG19)	84.10%

**Table 8. Accuracy on the Caltech 256 dataset with 60 training images per category.**

## 6. 结论

本文介绍了一种无监督的关于细粒度和普通对象分类的普通部件选择方法。给定一个预训练 CNN 网络来分类，我们利用学习获得的部件探测器作为普通部件探测器。通过分析预测所哟有训练图像部件位置来评估一个part constellation模型。最终模型包络一个有用的部件提案选择以及感兴趣对象在不同视角中的空间关系。

我们使用这种部件模型用于基于部件图像分类的细粒度识别和普通对象识别。和许多目前的细粒度工作相比，我们的方法优于这个领域最好的方法，它也有利于其它的任务，例如数据量增加和普通对象分类。有支持表明，在CUB200-2011的识别率是81.0%，在Caltech256的正确率是84.1%。

在我们未来的工作中，我们计划直接使用深度神经激活地图作为概率地图，同时保持我们方法的速度。对象规模的估计将允许应用我们的方法到一种图像数据集，在这些图像中只包含一小部件对象。我们目前的限制是一个对象部件对应一个单通道的假设。我们认为通道的结合可以提高定位的准确率。另外，我们计划在一个共同的框架下学习the constellation models 和随后的分类。