Selective Search for Object Recognition

J.R.R. Uijlings[1,2], K.E.A. van de Sande[2], T. Gevers[2], and A.W.M. Smeulders[2]

University of Trento, Italy
[2]University of Amsterdam, the Netherlands

Technical Report 2012, submitted to IJCV

摘要

本文探讨了在物体识别中生成可能物体位置的问题。我们引入了选择性搜索，它结合了穷举搜索和分割的优点。与分割一样，我们使用图像结构来指导采样过程。与穷举搜索一样，我们的目标是捕捉所有可能的物体位置。我们不使用单一技术来生成可能的物体位置，而是进行多样化搜索，使用各种互补的图像分割来处理尽可能多的图像条件。我们的 "选择性搜索 "产生了一小批数据驱动、与类别无关的高质量位置，在 10,097 个位置上的召回率为 99%，平均最佳重叠率为 0.879。与穷举式搜索相比，地点数量的减少使我们能够使用更强的机器学习技术和更强的外观模型来识别物体。在本文中，我们展示了我们的选择性搜索能够使用强大的词袋模型进行识别。选择性搜索软件已公开发布 。

1 引言

长期以来，人们一直寻求在识别物体之前对其进行划分。这就产生了分割，其目的是通过通用算法对图像进行独特的分割，即图像中的所有物体轮廓都有一个部分。在过去的几年中，有关这一主题的研究取得了巨大的进展[3, 6, 13, 26]。但是，图像本质上是分层的：在图 1a 中，沙拉和勺子位于沙拉碗中，而沙拉碗又位于桌子上。此外，根据语境的不同，图片中的桌子一词可以仅指木头，也可以包括桌子上的所有东西。因此，图像的性质和对象类别的不同用途都是有层次的。除了最特殊的用途外，这就禁止了对对象进行独特的划分。因此，对于大多数任务来说，分割中的多个尺度是必要的。最自然的解决方法就是使用分层分割，例如 Arbelaez 等人的研究[3]。

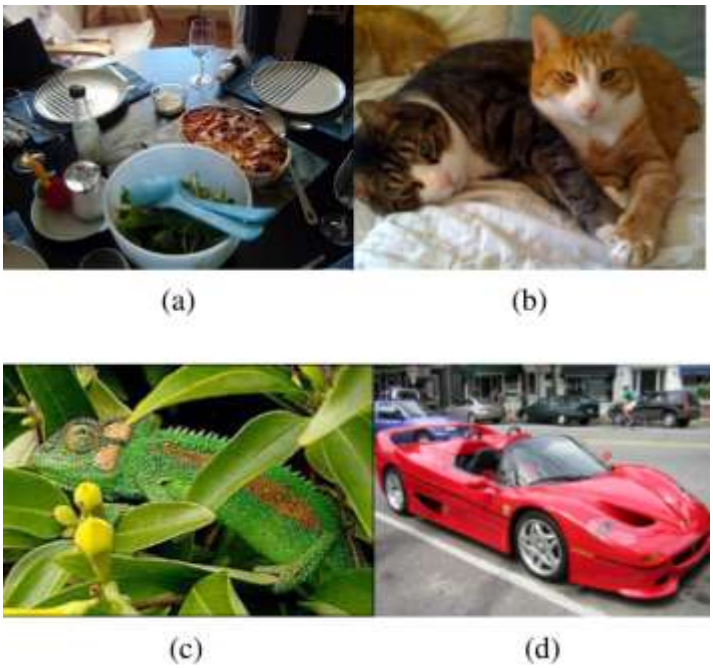

图 1：图像区域形成物体的原因多种多样。在(b)中，猫可以通过颜色而不是纹理来区分。在（c）中，变色龙可以通过纹理而不是颜色与周围的树叶区分开来。在(d)中，车轮之所以能成为汽车的一部分，是因为它们是封闭的，而不是因为它们的质地或颜色相似。因此，要想有条理地找到物体，就必须使用各种不同的策略。此外，图像在本质上是有层次的，因为在（a）中，没有一个单一的比例尺可以找到完整的桌子、沙拉碗和沙拉勺。

除了细分应具有层次性之外，使用单一策略进行细分的通用解决方案可能根本不存在。有很多相互矛盾的原因会导致一个区域被组合在一起：在图 1b 中，猫可以用颜色分开，但它们的纹理是一样的。相反，在图 1c 中，变色龙与周围的树叶颜色相似，但纹理却不同。最后，在图 1d 中，车轮在

颜色和纹理上都与汽车大相径庭，但却被汽车包围着。因此，单个视觉特征无法解决分割的模糊性问题。

最后，还有一个更根本的问题。只有在确定手头的物体是一个人之后，才能将具有截然不同特征的区域（如人脸和毛衣）合并为一个物体。因此，如果没有事先识别，就很难确定人脸和毛衣是一个物体的一部分 [29]。这就导致了与传统方法相反的做法：通过识别物体来进行定位。这种最新的物体识别方法在不到十年的时间里取得了巨大进步 [8, 12, 16, 35]。利用从示例中学习到的外观模型，进行穷举搜索，检查图像中的每一个位置，以免遗漏任何潜在的物体位置 [8、12、16、35]。

然而，穷举搜索本身有几个缺点。搜索所有可能的位置在计算上是不可行的。必须使用规则网格、固定比例和固定长宽比来缩小搜索空间。在大多数情况下，需要访问的位置数量仍然很大，以至于需要施加其他限制。分类器需要简化，外观模型需要快速。此外，均匀采样会产生许多方框，而这些方框显然不支持某个物体。与其盲目地使用穷举搜索进行位置采样，一个关键问题是：我们能否通过数据驱动分析来引导采样？

本文旨在结合分割和穷举搜索的最佳直觉，提出一种数据驱动的选择性搜索。受自下而上分割法的启发，我们旨在利用图像的结构来生成物体的位置。受穷举搜索的启发，我们的目标是捕捉所有可能的物体位置。因此，我们不使用单一的采样技术，而是采用多样化的采样技术，以尽可能多地考虑图像条件。具体来说，我们采用基于数据驱动的分组策略，通过使用各种互补分组标准和各种具有不同不变性的互补色彩空间来增加多样性。位置集是由这些互补分区的位置组合而成的。我们的目标是生成一种独立于类别、数据驱动的选择性搜索策略，从而生成一小部分高质量的物体位置。

我们选择性搜索的应用领域是物体识别。因此，我们在最常用的数据集上进行了评估，即帕斯卡尔 VOC 检测挑战，其中包括 20 个物体类别。该数据集的规模为我们的选择性搜索带来了计算上的限制。此外，使用该数据集意味着主要从边界框的角度来评估位置的质量。不过，我们的选择性搜索同样适用于区域，也适用于 "草 "等概念。

在本文中，我们提出了用于物体识别的选择性搜索。我们的主要研究问题是(1) 将分割作为选择性搜索策略时，有哪些好的多样化策略？(2) 选择性搜索在图像中创建一小部分高质量位置的效果如何？(3) 我们能否利用选择性搜索来使用更强大的分类器和外观模型来识别物体？

2 相关工作

我们将相关工作局限于物体识别领域，并将其分为三类：穷举搜索、分割和其他不属于这两类的采样策略。

2.1 穷举搜索

由于一个物体可以位于图像中的任何位置和比例，因此很自然地就会在所有地方进行搜索 [8, 16, 36]。然而，视觉搜索空间非常大，因此穷举搜索的计算成本很高。这就对每个位置的评估成本和/或考虑的位置数量造成了限制。因此，这些滑动窗口技术大多使用粗搜索网格和固定长宽比，使用弱分类器和经济图像特征，如 HOG [8, 16, 36]。这种方法通常用作级联分类器的预选步骤 [16, 36]。

与滑动窗口技术相关的是 Felzenszwalb 等人 [12] 非常成功的基于部件的物体定位方法。他们的方法也是使用线性 SVM 和 HOG 特征进行穷举搜索。不过，他们搜索的是物体和物体的部分，这两者的结合产生了令人印象深刻的物体检测性能。

Lampert 等人[17] 建议使用外观模型来指导搜索。这既减轻了使用规则网格、固定比例和固定长宽比的限制，同时又减少了访问位置的数量。这是通过使用分支和边界技术在图像中直接搜索最佳窗口来实现的。虽然他们为线性分类器取得了令人印象深刻的结果，但 [1] 发现，对于非线性分类器，该方法在实践中仍会访问每幅图像中超过 100,000 个窗口。

我们提出了选择性搜索，而不是盲目的穷举搜索或分支搜索。我们使用底层图像结构来生成对象位置。与已讨论过的方法不同的是，这种方法能生成完全独立于类别的位置集。此外，由于我们没有使用固定的长宽比，因此我们的方法并不局限于物体，还能找到 "草 "和 "沙 "等物体（这一点在 [17] 中也是成立的）。最后，我们希望能生成更少的位置，这将使问题变得更容易，因为样本的可变性变

图 2：我们选择性搜索的两个例子，显示了不同尺度的必要性。左图中，我们发现了许多不同比例的物体。在右图中，由于女孩被 TV 所包含，我们必然会在不同尺度上找到物体。

低了。更重要的是，这样可以释放计算能力，用于更强的机器学习技术和更强大的外观模型。

2.2 分类

Carreira 和 Sminchisescu [4] 以及 Endres 和 Hoiem [9] 都建议利用分割生成一组与类别无关的物体假设。这两种方法都会生成多个前景/背景分割，学习预测前景分割是一个完整物体的可能性，并以此对分割进行排序。这两种算法都显示出了在图像中准确划分物体的能力，这一点得到了 [19] 的证实，他们在使用 [4] 进行像素图像分类时取得了最先进的结果。与常见的分割方法一样，这两种方法都依赖于一种单一的强大算法来识别良好区域。它们通过使用许多随机初始化的前景和背景种子来获得各种位置。相比之下，我们通过使用不同的分组标准和不同的表示方法来明确处理各种图像条件。这意味着计算投资更低，因为我们不必投资于单一的最佳分割策略，例如使用 [3] 出色但昂贵的轮廓检测器。此外，由于我们分别处理不同的图像条件，因此我们希望我们的定位质量更加一致。最后，我们的选择性搜索范式决定了最有趣的问题不是我们的区域与 [4, 9] 相比如何，而是它们如何能够相互补充。

Gu 等人[15]解决了根据物体的部件仔细分割和识别物体的问题。他们首先使用基于 Arbelaez 等人 [3] 的分组方法生成一组部件假设。每个部件假设都由外观和形状特征描述。然后，利用物体的各个部分对其进行识别和仔细划分，从而获得良好的形状识别效果。在他们的工作中，分割是分层的，并产生所有尺度上的分割。不过，他们使用了单一的分组策略，其发现部分或物体的能力没有得到评估。在这项工作中，我们使用多种互补策略来处理尽可能多的图像条件。我们将使用 [3] 所生成的位置也纳入了评估范围。

2.3 其他取样策略

阿列克谢等人[2]为解决穷举搜索采样空间大的问题，提出了搜索任何物体（与其类别无关）的方法。在他们的方法中，他们在具有明确形状的物体（而不是 "草 "和 "沙 "之类的物体）的物体窗口上训练分类器。然后，他们不进行全面的穷举搜索，而是随机抽取盒子，并将分类器应用于这些盒子。具有最高 "对象性 "度量的方框将作为一组对象假设。然后，这组假设被用来大大减少特定类别对象检测器所评估的窗口数量。我们将我们的方法与他们的工作进行比较。

另一种策略是使用词袋模型中的视觉词来预测物体位置。Vedaldi 等人[34]使用跳跃窗口[5]，通过学习单个视觉词与物体位置之间的关系来预测新图像中的物体位置。Maji 和 Malik[23]结合了这些关系中的多个关系，使用 Hough 变换来预测物体位置，然后他们随机采样接近 Hough 最大值的窗口。与学习不同的是，我们利用图像结构来采样一组与类别无关的物体假设。

总之，我们的创新点如下。我们不采用穷举式搜索 [8, 12, 16, 36]，而是使用分割作为选择性搜索，从而获得一小部分与类别无关的物体位置。与 [4, 9] 的分割方法不同，我们并不专注于最佳分割算法 [3]，而是使用多种策略来处理尽可能多的图像条

件，从而大大降低了计算成本，同时有可能准确捕捉到更多的物体。我们不是在随机抽样的方框上学习对象度量[2]，而是使用自下而上的分组程序来生成良好的对象位置。

3 选择性搜索

在本节中，我们将详细介绍用于物体识别的选择性搜索算法，并介绍各种多样化策略，以应对尽可能多的图像条件。选择性搜索算法在设计时需要考虑以下因素：

多样化：将区域组合在一起并没有单一的最佳策略。如图 1 所示，各区域组成一个物体的原因可能只是颜色、纹理或部分区域被包围。此外，阴影和光线颜色等照明条件也会影响区域形成物体的方式。因此，我们希望有一套多样化的策略来应对所有情况，而不是在大多数情况下都采用单一的策略。

快速计算：选择性搜索的目标是生成一组可能的物体位置，供实际物体识别框架使用。创建这组位置不应该成为计算瓶颈，因此我们的算法应该相当快速。

3.1 通过分层分组进行选择性搜索

我们采用分层分组算法作为选择性搜索的基础。自下而上的分组是一种常用的分割方法 [6, 13]，因此我们将其用于选择性搜索。由于分组过程本身是分层的，我们可以通过继续分组过程自然生成所有尺度的位置，直到整个图像成为一个单一区域。这就满足了捕捉所有尺度的条件。

由于区域比像素能提供更丰富的信息，我们希望尽可能使用基于区域的特征。为了获得一组理想情况下不会跨越多个对象的小型起始区域，我们使用了 Felzenszwalb 和 Huttenlocher [13] 的快速方法。

我们现在的分组程序如下。我们首先使用 [13] 创建初始区域。然后，我们使用贪婪算法对区域进行迭代分组：首先计算所有相邻区域之间的相似度。首先计算所有相邻区域之间的相似度，然后将两个最相似的区域组合在一起，并计算由此产生的区域与其相邻区域之间新的相似度。重复将最相似区域分组的过程，直到整个图像成为一个单独的区域。一般方法详见算法 1。

Algorithm 1: Hierarchical Grouping Algorithm

Input: (colour) image
Output: Set of object location hypotheses L

Obtain initial regions $R = \{r_1, \cdots, r_n\}$ using [13]
Initialise similarity set $S = \emptyset$
foreach *Neighbouring region pair* (r_i, r_j) **do**
  Calculate similarity $s(r_i, r_j)$
  $S = S \cup s(r_i, r_j)$

while $S \neq \emptyset$ **do**
  Get highest similarity $s(r_i, r_j) = \max(S)$
  Merge corresponding regions $r_t = r_i \cup r_j$
  Remove similarities regarding $r_i : S = S \setminus s(r_i, r_*)$
  Remove similarities regarding $r_j : S = S \setminus s(r_*, r_j)$
  Calculate similarity set S_t between r_t and its neighbours
  $S = S \cup S_t$
  $R = R \cup r_t$

Extract object location boxes L from all regions in R

对于区域 ri 和区域 rj 之间的相似度 s(ri,rj)，我们希望有多种互补的测量方法，但这些方法必须能够快速计算。实际上，这意味着相似度应基于可在层次结构中传播的特征，即在将区域 ri 和 rj 合并为区域 rt 时，需要根据区域 ri 和 rj 的特征计算区域 rt 的特征，而无需访问图像像素。

3.2 多样化战略

选择性搜索的第二个设计标准是使采样多样化，并创建一组互补策略，然后将其位置进行组合。我们将选择性搜索多样化：(1) 使用具有不同不变性的各种颜色空间；(2) 使用不同的相似性度量 sij；(3) 改变起始区域。

互补色空间：我们希望考虑到不同的场景和照明条件。因此，我们在各种具有不同不变性的色彩空间中执行分层分组算法。具体来说，我们采用了以下不变性程度依次递增的色彩空间：(1) RGB，(2) 强度（灰度图像）I，(3) Lab，(4) 归一化 RGB 加强度的 rg 通道（表示为 rgI），(5) HSV，(6) 归一化 RGB（表示为 rgb），(7) C [14]，这是一个强度被分割出来的对立色彩空间，最后 (8) HSV 中的色调通道 H。表 1 列出了具体的不变性属性。

当然，对于黑白图像，改变色彩空间对算法的最终结果影响不大。对于这些图像，我们依靠其他多样化方法来确保良好的物体定位。

在本文中，我们在整个算法中始终使用单一的色彩空间，也就是说，[13] 的初始分组算法和我们随后的分组算法都是在这个色彩空间中执行的。互补相似性度量。我们定义了四个互补的、快速计

colour channels	R	G	B	I	V	L	a	b	S	r	g	C	H
Light Intensity	-	-	-	-	-	-	+/-	+/-	+	+	+	+	+
Shadows/shading	-	-	-	-	-	-	+/-	+/-	+	+	+	+	+
Highlights	-	-	-	-	-	-	-	-	-	-	-	+/-	+

表 1：本文使用的各个色彩通道和色彩空间的不变性属性，按不变性程度排序。+/-"表示部分不变。分数 1/3 表示三个色彩通道中的一个具有上述不变性。

算的相似性度量。这些度量的范围都是 [0,1]，这就为这些度量的组合提供了便利。

scolour(ri,rj) 测量色彩相似性。具体来说，对于每个区域，我们使用 25 个分区获得每个颜色通道的一维颜色直方图，我们发现这种方法效果很好这就为每个区域 ri 得出了色彩直方图 $C_i = \{c_i^1, \cdots, c_i^n\}$，当使用三个色彩通道时，维数为 n = 75。颜色直方图使用 L1 规范进行归一化处理。相似度通过直方图交集来衡量：

$$s_{colour}(r_i, r_j) = \sum_{k=1}^{n} \min(c_i^k, c_j^k). \quad (1)$$

颜色直方图可以通过以下方式在层次结构中有效传播：

$$C_t = \frac{\text{size}(r_i) \times C_i + \text{size}(r_j) \times C_j}{\text{size}(r_i) + \text{size}(\text{r}_\text{j})}. \quad (2)$$

结果区域的大小就是其组成部分的总和：size(rt) = size(ri)+size(rj) 。

stexture(ri,rj) 测量纹理相似性。我们使用类似于 SIFT 的快速测量方法来表示纹理，因为 SIFT 本身在材料识别方面效果很好 [20]。我们对每个颜色通道的八个方向使用 σ = 1 取高斯导数。对于每个颜色通道的每个方向，我们使用 10 个二进制大小来提取直方图。这样就为每个区域 ri 得出了纹理直方图 Ti = {ti1,--- ,tin}，当使用三个颜色通道时，维数为 n = 240。纹理直方图使用 L1 规范进行归一化处理。相似度通过直方图交集进行测量：

$$s_{texture}(r_i, r_j) = \sum_{k=1}^{n} \min(t_i^k, t_j^k). \quad (3)$$

纹理直方图与颜色直方图一样，可以通过层次结构有效传播。

Ssize(ri,rj) 鼓励小区域尽早合并。这就迫使 S 中的区域（即尚未合并的区域）在整个算法过程中大小相似。这样做是可取的，因为它能确保在图像的所有部分都能创建各种尺度的对象位置。例如，它可以防止单个区域逐个吞噬其他所有区域，导致所有尺度的对象只出现在这个增长区域的位置上，而其他地方都没有。size(ri,rj) 被定义为 ri 和 rj 共同占据图像的部分：

$$s_{size}(r_i, r_j) = 1 - \frac{\text{size}(r_i) + \text{size}(\text{r}_\text{j})}{\text{size}(im)}, \quad (4)$$

其中，size(im) 表示图像的像素大小。

sfill(ri,rj) 测量区域 ri 和区域 rj 的相互匹配程度。其目的是填补空白：如果 ri 包含在 rj 中，那么理应先合并这两个区域，以避免出现任何空洞。另一方面，如果 ri 和 rj 几乎没有相互接触，它们很可能会形成一个奇怪的区域，因此不应合并。为了保持测量的快速性，我们只使用区域和包含方框的大小。具体来说，我们将 BBij 定义为 ri 和 rj 周围的紧密包围盒。现在，sfill(ri,rj) 是 BBij 所包含的图像中未被 ri 和 rj 区域覆盖的部分：

$$fill(r_i, r_j) = 1 - \frac{\text{size}(BB_{ij}) - \text{size}(r_i) - \text{size}(r_i)}{\text{size}(im)} \quad (5)$$

为了与等式 4 保持一致，我们将其除以 size(im)。请注意，通过跟踪每个区域周围的边界框可以有效地计算出这一指标，因为两个区域周围的边界框可以很容易地从中推导出来。

在本文中，我们最终采用的相似度测量方法是上述四种方法的组合：

$$s(r_i, r_j) = a_1 s_{colour}(r_i, r_j) + a_2 s_{texture}(r_i, r_j) + a_3 s_{size}(r_i, r_j) + a_4 s_{fill}(r_i, r_j), \quad (6)$$

其中，ai ∈ {0,1} 表示是否使用了相似性度量。由于我们的目标是使我们的策略多样化，因此我们不考虑任何加权相似性。

互补起始区域：第三种多样化策略是改变互补起始区域。据我们所知，[13] 的方法是最快的、公开可用的算法，能产生高质量的起始位置。我们找不到其他具有类似计算效率的算法，因此本文只使用这种过度分割法。但需要注意的是，不同的起始区域（已经）可以通过改变色彩空间来获得，而每种色彩空间都具有不同的不变性。此外，我们还改变了 [13] 中的阈值参数 k。

3.3 组合位置

在本文中，我们将结合分层分组算法的几种变体来组合对象假设。在理想情况下，我们希望对对象假设进行排序，使最有可能成为对象的位置排在

前面。这样，我们就能根据后续特征提取和分类方法的计算效率，在所得到的对象假设集的质量和数量之间找到一个很好的平衡点。

我们选择根据每个分组策略中生成假设的顺序来排列组合对象假设集。然而，由于我们合并了多达 80 种不同策略的结果，这样的排序会过于强调大区域。为了避免这种情况，我们加入了以下随机性。给定一个分组策略 j，让 rij 成为在层次结构中第 i 个位置创建的区域，其中 i = 1 代表层次结构的顶端（其对应区域覆盖整个图像）。现在我们计算位置值 vij，即 RND×i，其中 RND 是范围为 [0,1] 的随机数。利用 vij 对区域进行排序，就得到了最终排名。

当我们使用边界框中的位置时，我们首先对所有位置进行排序，详见上文。之后，我们才会过滤掉排名较低的重复位置。这样可以确保重复的方框有更大的机会获得较高的排名。这样做是可取的，因为如果多个分组策略建议使用相同的方框位置，那么它很可能来自图像中视觉上一致的部分。

4 利用选择性搜索识别物体

本文利用选择性搜索生成的位置进行物体识别。本节将详细介绍我们的目标识别框架。

有两种特征在物体识别中占主导地位：定向梯度直方图（HOG）[8] 和词袋[7, 27]。Felzenszwalb 等人[12]的研究表明，将 HOG 与基于部分的模型相结合是成功的。然而，由于他们使用的是穷举搜索，从计算角度来看，HOG 特征与线性分类器的结合是唯一可行的选择。相比之下，我们的选择性搜索可以使用更昂贵、潜在功能更强大的特征。因此，我们使用词袋进行物体识别 [16, 17, 34]。不过，我们采用了多种颜色-SIFT 描述子[32]和更精细的空间金字塔划分[18]，从而实现了比[16, 17, 34]更强大（也更昂贵）的功能。

具体来说，我们对每个像素的描述符进行单比例采样（ σ = 1.2）。使用 [32] 中的软件，我们提取了 SIFT [21] 和两种彩色 SIFT（Extended OpponentSIFT [31] 和 RGBSIFT [32]），这两种 SIFT 被认为对检测图像结构最为敏感。我们使用大小为 4,000 的视觉编码本和一个有 4 个层次的空间金字塔，使用 1x1、2x2、3x3 和 4x4 的划分。这样，总的特征向量长度为 360,000 个。在图像分类中，已经使用过这种大小的特征[25, 37]。由于空间金字塔产生的空间细分比构成 HOG 描述子的单元更粗，因此我们的特征包含的有关物体特定空间布局的信息更少。因此，HOG 更适用于刚性物体，而我们的特征更适用于可变形物体类型。

作为分类器，我们使用 Shogun 工具箱[28]，使用带有直方图交集核的支持向量机。为了应用训练有素的分类器，我们使用了 [22] 的快速近似分类策略，该策略在 [30] 中被证明在词袋分类中效果良好。

我们的训练过程如图 3 所示。初始正向示例包括所有基本真实对象窗口。作为初始负示例，我们从选择性搜索生成的所有对象位置中选出与正示例重叠 20% 到 50% 的位置。为了避免近乎重复的负示例，如果一个负示例与另一个负示例重叠超过 70%，就会被排除在外。为了将每个类别的初始负例数量控制在 20,000 个以下，我们随机放弃了汽车、猫、狗和人类别中一半的负例。直观地说，这组例子可以看作是与正面例子相近的困难底片。这意味着它们接近决策边界，因此，即使考虑到完整的负面例子集，它们也有可能成为支持向量。事实上，我们发现这种训练示例的选择可以提供相当不错的初始分类模型。

然后，我们进入再训练阶段，迭代地添加硬负面示例（例如 [12]）：我们使用选择性搜索生成的位置，将学习到的模型应用于训练集。对于每个负面图像，我们都会添加得分最高的位置。由于初始训练集已经产生了良好的模型，我们的模型只需两次迭代即可收敛。

对于测试集，最终模型应用于选择性搜索生成的所有位置。窗口按分类器得分排序，而与得分较高的窗口重叠超过 30% 的窗口则被视为近乎重复的窗口并被删除。

5 评估

在本节中，我们将对选择性搜索的质量进行评估。我们将实验分为四个部分，每个部分都有一个独立的小节：

多样化策略：我们尝试了多种颜色空间、相似度测量方法和初始区域的阈值，所有这些在第 3.2 节中都有详细介绍。我们力求在生成的对象假设数量、计算时间和对象位置质量之间取得平衡。我们

通过边界框来实现这一点。这样，我们选择了多种互补技术，并将其作为最终的选择性搜索方法。

定位质量：我们对选择性搜索得出的物体位置假设的质量进行测试。

物体识别：我们在第 4 节详述的物体识别框架中使用了选择性搜索的位置。我们在 Pascal VOC 检测挑战中对性能进行了评估。

位置质量的上限：我们研究了我们的物体识别框架在使用 "完美 "质量的物体假设集时的表现。这与我们的选择性搜索所生成的位置相比如何？

为了评估对象假设的质量，我们定义了平均最佳重叠度 (ABO) 和平均最佳重叠度 (MABO) 分数，这是对 [9] 中使用的测量方法的略微概括。为了计算特定类别 c 的平均最佳重叠度，我们计算了每个地面实况注释 gci ∈ Gc 与为相应图像生成的对象假设 L 之间的最佳重叠度，并求取平均值：

$$\mathrm{ABO} = \frac{1}{|G^c|} \sum_{g_i^c \in G^c} \max_{l_j \in L} \mathrm{Overlap}(g_i^c, l_j). \tag{7}$$

重叠分值取自文献 [11]，测量两个区域的交集面积除以其合并面积：

$$\mathrm{Overlap}(g_i^c, l_j) = \frac{\mathrm{area}(g_i^c) \cap \mathrm{area}(\mathrm{l_j})}{\mathrm{area}(g_i^c) \cup \mathrm{area}(\mathrm{l_j})}. \tag{8}$$

与平均精度和平均平均精度类似，平均最佳重叠度现在定义为所有类别的平均 ABO。

其他研究通常使用帕斯卡重叠标准（Pascal Overlap Criterion）得出的召回率来衡量方框的质量 [1, 16, 34]。该标准认为，当等式 8 中的重叠度大于 0.5 时，物体就被找到了。不过，在我们的许多实验中，大多数类别的召回率都在 95% 到 100% 之间，因此对于本文来说，这一标准过于不敏感。不过，在与其他工作进行比较时，我们还是会报告这一指标。

为避免过度拟合，我们在 Pascal VOC 2007 TRAIN+VAL 集上进行了多样化策略实验。其他实验在 Pascal VOC 2007 TEST 集上进行。此外，我们还使用独立评估服务器，在 Pascal VOC 2010 检测挑战赛中对我们的物体识别系统进行了基准测试。

5.1 多样化战略

在本节中，我们将对各种策略进行评估，以便在合理的时间内使用合理数量的方框计算获得高质量的物体位置假设。

5.1.1 扁平化与层次化

在我们的方法描述中，我们声称使用完整的层次结构比通过改变阈值使用多个平面分区更自然。在本节中，我们将测试使用层次结构是否也能带来更好的结果。因此，我们将使用多个阈值的 [13] 算法与我们提出的算法进行比较。具体来说，我们在 RGB 色彩空间中执行这两种策略。对于 [13]，我们将阈值以 50 为单位从 k = 50 到 k = 1000 变化。这个范围既包括小区域，也包括大区域。此外，作为一种特殊的阈值类型，我们将整幅图像作为对象位置，因为很多图像只包含一个大型对象。此外，我们还采用了一个更粗的范围，从 k = 50 到 k = 950，每 100 步为一个阈值。在我们的算法中，我们使用 k = 50 的阈值来创建初始区域，以确保两种策略具有相同的最小尺度。此外，由于我们生成的区域较少，我们将使用 k = 50 和 k = 100 的结果合并在一起。我们使用公式 6 中定义的所有四个相似度的加和作为相似度量 S。结果见表 2。

threshold k in [13]	MABO	# windows
Flat [13] $k = 50, 150, \cdots, 950$	0.659	387
Hierarchical (this paper) $k = 50$	0.676	395
Flat [13] $k = 50, 100, \cdots, 1000$	0.673	597
Hierarchical (this paper) $k = 50, 100$	0.719	625

表 2：多平面分区与分层分区在生成方框位置方面的比较显示，分层策略的平均最佳重叠 (MABO) 得分在位置数量相似的情况下始终较高。

可以看出，我们的分层策略比多重平面分割策略的对象假设质量更高：在区域数量相似的情况下，我们的 MABO 得分一直较高。此外，将我们的分层分组算法的两个变体的位置结合在一起所实现的 MABO 增加值要远远高于为平面分割添加额外阈值所实现的增加值。我们的结论是，使用分层分组算法的所有位置不仅更自然，而且比使用多个平面分区更有效。

5.1.2 单个多样化策略

本文提出了三种多样化策略来获得高质量的对象假设：改变色彩空间、改变相似性度量以及改变获得起始区域的阈值。本节将研究每种策略的影响。

Similarities	MABO	# box
C	0.635	356
T	0.581	303
S	0.640	466
F	0.634	449
C+T	0.635	346
C+S	0.660	383
C+F	0.660	389
T+S	0.650	406
T+F	0.638	400
S+F	0.638	449
C+T+S	0.662	377
C+T+F	0.659	381
C+S+F	0.674	401
T+S+F	0.655	427
C+T+S+F	0.676	395

Colours	MABO	# box
HSV	0.693	463
I	0.670	399
RGB	0.676	395
rgI	0.693	362
Lab	0.690	328
H	0.644	322
rgb	0.647	207
C	0.615	125

Thresholds	MABO	# box
50	0.676	395
100	0.671	239
150	0.668	168
250	0.647	102
500	0.585	46
1000	0.477	19

表 3：使用各种分割策略的基于盒状物体假设的平均最佳重叠率。(C)olour、(S)ize 和 (F)ill 的表现类似。(T)exture 本身较弱。最佳组合是尽可能多的不同来源。

作为基本设置，我们使用 RGB 色彩空间、所有四种相似性测量方法的组合以及阈值 k = 50。每次我们只改变一个参数。结果见表 3。

我们从表 3 左侧的相似性度量组合开始研究。首先分别查看颜色、纹理、大小和填充，我们发现纹理相似性表现最差，MABO 为 0.581，而其他测量值在 0.63 和 0.64 之间。为了测试纹理得分相对较低是否与我们选择的特征有关，我们还尝试用局部二进制模式来表示纹理[24]。我们尝试在不同尺度上使用 4 个和 8 个邻域，并使用不同的图案均匀度/一致性（见 [24]），我们将各个颜色通道的 LBP 直方图串联起来。然而，我们得到了相似的结果（MABO 为 0.577）。我们认为，纹理的弱点之一是物体边界：当两个片段被物体边界分开时，边界两侧会产生相似的边缘响应，从而无意中增加了相似性。

虽然纹理相似性得出的物体位置相对较少，但在 300 个位置时，其他相似性测量的 MABO 仍高于 0.628。这表明，在比较单个策略时，表 3 中的最终 MABO 分数是权衡物体假设质量和数量的良好指标。另一个观察结果是，相似性度量的组合通常优于单一度量。事实上，使用所有四种相似性测量方法的效果最好，MABO 为 0.676。

从表 3 右上方的色彩空间变化来看，我们发现结果差异很大，C 色彩空间 125 个位置的 MABO 为 0.615，而 HSV 色彩空间 463 个位置的 MABO 为 0.693。我们注意到，Lab 空间仅用 328 个方框就获得了 0.690 的 MABO 高分。此外，每个层次的顺序

Version	Diversification Strategies	MABO	# win	# strategies	time (s)
Single Strategy	HSV C+T+S+F $k = 100$	0.693	362	1	0.71
Selective Search Fast	HSV, Lab C+T+S+F, T+S+F $k = 50, 100$	0.799	2147	8	3.79
Selective Search Quality	HSV, Lab, rgI, H, I C+T+S+F, T+S+F, F, S $k = 50, 100, 150, 300$	0.878	10,108	80	17.15

表 4：通过贪婪搜索得出的选择性搜索方法。我们将所选的各个分散策略进行组合，得出分层分组算法的 1、8 和 80 种变体。随着窗口数量的增加，平均最佳重叠度（MABO）得分也在稳步上升。

method	recall	MABO	# windows
Arbelaez *et al.* [3]	0.752	0.649 ± 0.193	418
Alexe *et al.* [2]	0.944	0.694 ± 0.111	1,853
Harzallah *et al.* [16]	0.830	-	200 per class
Carreira and Sminchisescu [4]	0.879	0.770 ± 0.084	517
Endres and Hoiem [9]	0.912	0.791 ± 0.082	790
Felzenszwalb *et al.* [12]	0.933	0.829 ± 0.052	100,352 per class
Vedaldi *et al.* [34]	0.940	-	10,000 per class
Single Strategy	0.840	0.690 ± 0.171	289
Selective search "Fast"	0.980	0.804 ± 0.046	2,134
Selective search "Quality"	0.991	0.879 ± 0.039	10,097

表 5：各种方法在 Pascal 2007 测试集上的召回率、平均最佳重叠率 (MABO) 和窗口位置数的比较。

也很有效：使用 HSV 色彩空间的前 328 个方框可获得 0.690 的 MABO 分数，而使用前 100 个方框可获得 0.647 的 MABO 分数。这表明，在比较单一策略时，我们可以只使用 MABO 分数来表示对象假设集的质量和数量之间的权衡。在下一节中，我们将利用这一点来寻找好的组合。

表 3 右下方对 [13] 生成起始区域的阈值进行了实验，结果表明，初始阈值越低，使用的对象位置越多，MABO 值越高。

5.1.3 多样化战略组合

我们使用多种互补的分组策略来组合物体位置假设，以获得一组高质量的物体位置。由于全面搜索最佳组合的计算成本很高，因此我们只使用 MABO 分数作为优化标准，进行贪婪搜索。我们在前面已经观察到，这个分数代表了位置数量和位置质量之间的权衡。

根据排序结果，我们创建了三种配置：单一最佳策略、快速选择性搜索和质量选择性搜索，其中质量选择性搜索使用了各个组成部分的所有组合，即色彩空间、相似度和阈值，详见表 4。贪婪搜索强调相似性测量组合的变化。这证实了我们的多样化假设：在高质量版本中，除了所有相似度的组合

外，填充和大小也被单独考虑。本文其余部分采用表 4 中的三种策略。

5.2 定位质量

在本节中，我们将从平均最佳重叠率和 Pascal VOC 2007 测试集上的定位数量两个方面对我们的选择性搜索算法进行评估。我们首先评估基于方框的位置，然后简要评估基于区域的位置。

5.2.1 基于方框的定位

我们比较了[16]的滑动窗口搜索、[12]使用其模型的窗口比率的滑动窗口搜索、[34]的跳跃窗口、[2]的 "对象性 "方框、[3]的分层分割算法周围的方框、[9]的区域周围的方框和[4]的区域周围的方框。在这些算法中，只有[3]不是为寻找物体位置而设计的。然而，[3] 是目前公开的最好的轮廓检测器之一，并能得出自然的区域层次结构。我们将其纳入评估范围，看看这种为分割而设计的算法是否也能很好地找到目标位置。此外，[4, 9] 算法的设计目的是找到好的物体区域，而不是方框。结果如表 5 和图 4 所示。

如表 5 所示，我们的 "快速 "和 "优质 "选择性搜索方法的召回率分别为 98% 和 99%，接近最佳召回率。就 MABO 而言，我们分别达到了 0.804 和 0.879。为了理解 0.879 的最佳重叠度意味着什么，图 5 显示了自行车、奶牛和人的重叠度在 0.874 和 0.884 之间的位置示例。这说明，我们的选择性搜索可以得到高质量的目标位置。

此外，请注意我们的 MABO 分数的标准偏差相对较低：快速选择性搜索的标准偏差为 0.046，高质量选择性搜索的标准偏差为 0.039。这表明，选择性搜索对物体属性的差异以及通常与特定物体相关的图像条件（例如室内/室外照明）具有很强的鲁棒性。

与其他算法相比，召回率第二高的算法是跳窗算法[34]，每类使用 10,000 个方框，召回率为 0.940。由于我们没有准确的方框，因此无法获得 MABO 分数。紧随其后的是 [12] 的穷举搜索，该方法在每类 100,352 个方框（该数字为所有类别的平均值）的情况下，召回率为 0.933，MABO 为 0.829。这一结果明显低于我们的方法，而使用的对象位置至少比我们的方法多 10 倍。

此外，请注意 [4, 9] 的分割方法具有相对较高的标准偏差。这说明，单一策略不可能对所有类别都同样有效。相反，使用多种互补策略会带来更稳定、更可靠的结果。

如果将 Arbelaez [3] 的分割方法与我们方法的单一最佳策略进行比较，他们在 418 个方框中的召回率为 0.752，MABO 为 0.649，而我们在 286 个方框中的召回率为 0.875，MABO 为 0.698。这表明，好的分割算法并不会自动产生好的目标位置边界框。

图 4 探讨了对象假设的质量和数量之间的权衡。就召回率而言，我们的 "快速 "方法优于所有其他方法。对于他们使用的 200 个位置来说，[16] 的方法似乎很有竞争力，但在他们的方法中，方框的数量是按类计算的，而在我们的方法中，所有类别都使用相同的方框。就 MABO 而言，[4]和[9]的对象假设生成方法在生成每幅图像多达 790 个对象方框位置时，都能很好地权衡数量/质量。不过，这些算法的计算成本分别是我们 "快速 "方法的 114 倍和 59 倍。

有趣的是，[2] 的 "对象性 "方法在召回率方面表现很好，但在 MABO 方面却差很多。这很可能是由于他们采用了非最大压制法，即压制与现有的、排名更靠前的窗口有 0.5 分以上重叠的窗口。虽然这种方法在以 0.5 分的重叠率来定义寻找对象时能明显改善结果，但对于寻找最高质量位置的一般问题来说，这种策略就不那么有效了，甚至会因为忽略了更好的位置而造成危害。

图 6 显示了几种方法在每个类别中的平均最佳重叠率。从图中可以看出，在自行车、桌子、椅子和沙发等类别中，[12] 的穷举搜索方法使用了多 10 倍的特定类别位置，其表现与我们的方法相似，而在其他类别中，我们的方法获得了最佳分数。一般来说，得分最高的类别是猫、狗、马和沙发，这几个类别比较容易识别，主要是因为数据集中的实例往往很大。得分最低的类别是瓶子、人和植物，这些类别比较困难，因为实例往往很小。然而，牛、羊和电视并不比人大，但我们的算法却能很好地找到它们。

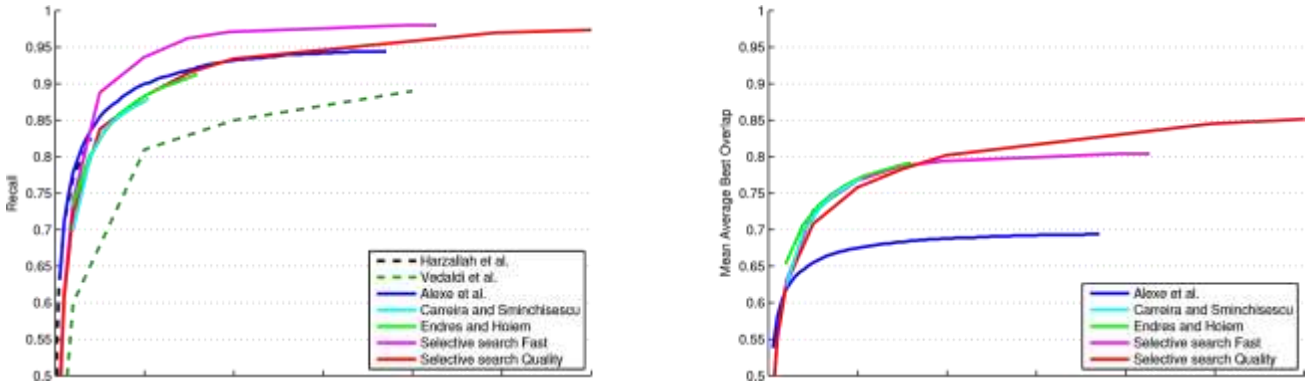

图 4：在 Pascal 2007 TEST 数据集上以边界框表示的对象假设的质量和数量之间的权衡。虚线表示的是数量以每类方框数量表示的方法。就召回率而言，"快速 "选择性搜索具有最佳权衡效果。就平均最佳重叠率而言，"优质 "选择性搜索与[4, 9]不相上下，但计算速度更快，持续时间更长，最终的平均最佳重叠率为 0.879。

(a)Bike :0.863 (b)Cow :0.874 (c)Chair :0.884 (d)Person :0.882 (e)Plant :0.873

图 5：最佳重叠度在平均最佳重叠度 0.879 左右的物体位置示例。绿色方框为地面实况。红色方框是使用 "质量 "选择性搜索创建的。

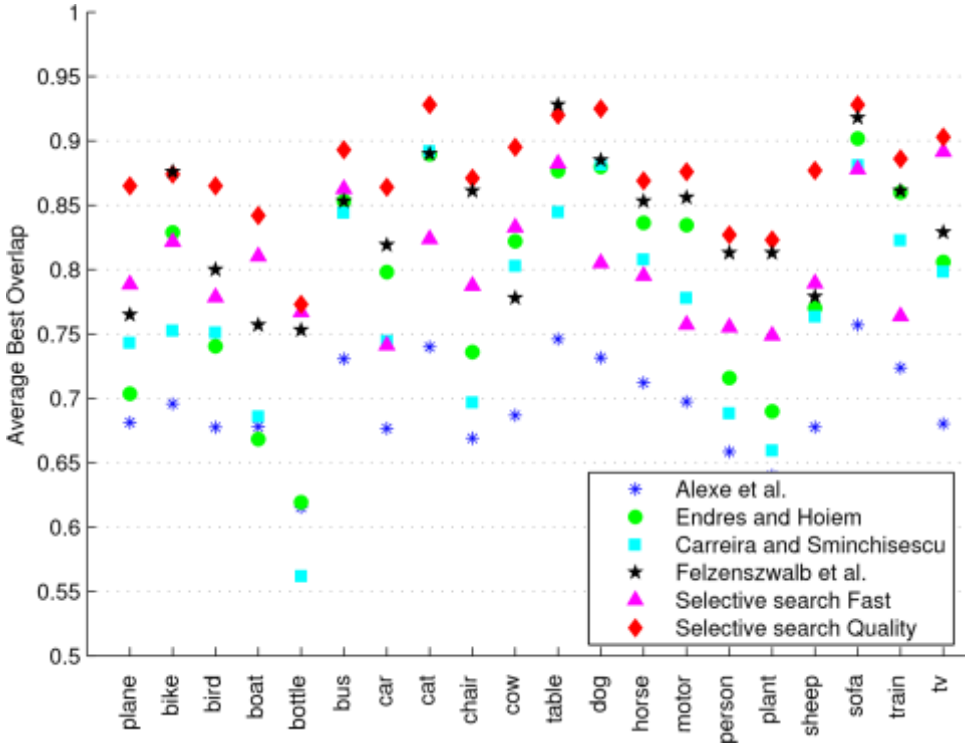

图 6：在 Pascal VOC 2007 TEST 上使用几种方法生成基于方框的对象位置时，每个类别的平均最佳重叠分数。对于除表格以外的所有类别，我们的 "高质量 "选择性搜索都能获得最佳位置。在 20 个类别中的 12 个类别中，我们的 "快速 "选择性搜索优于昂贵的 [4, 9]。我们始终优于 [2]。

method	recall	MABO	# regions	time(s)
[3]	0.539	0.540 ± 0.117	1122	64
[9]	0.813	0.679 ± 0.108	2167	226
[4]	0.782	0.665 ± 0.118	697	432
Single Strategy	0.576	0.548 ± 0.078	678	0.7
"Fast"	0.829	0.666 ± 0.089	3574	3.8
"Quality"	0.904	0.730 ± 0.093	22,491	17
[4, 9] + "Fast"	0.896	0.737 ± 0.098	6,438	662
[4, 9] + "Quality"	0.920	0.758 ± 0.096	25,355	675

表 6：在 Pascal 2007 TEST 的分割部分，比较各种算法如何在区域方面找到一组好的潜在对象位置。

总之，选择性搜索在使用有限数量的方框找到高质量的对象假设集方面非常有效，而且在对象类别中质量是合理一致的。文献[4]和[9]中的方法对多达 790 个对象位置的质量/数量权衡效果类似。不过，它们在对象类别上的差异更大。此外，与我们的 "快速 "和 "高质量 "选择性搜索方法相比，它们的计算成本分别至少高出 59 倍和 13 倍，这对于目前的物体识别数据集规模来说是个问题。总之，我们得出结论，选择性搜索能在 0.879 MABO 的条件下获得最佳质量的位置，同时使用合理数量的 10,097 个与类别无关的物体位置。

5.2.2 基于区域的定位

在本节中，我们将检验选择性搜索所生成的区域在捕捉物体位置方面的效果。我们在 Pascal VOC 2007 TEST 集的分割部分进行了这项研究。我们将其与 [3] 的分割以及 [4, 9] 的对象假设区域进行比较。表 6 显示了结果。请注意，由于几乎没有完全相同的区域，因此区域的数量大于方框的数量。

[4, 9]的目标区域质量与我们的 "快速 "选择性搜索相似，分别为 0.665 MABO 和 0.679 MABO，而我们的 "快速 "搜索结果为 0.666 MABO。虽然 [4, 9] 使用的区域较少，但这些算法的计算成本分别高出 114 倍和 59 倍。我们的 "高质量 "选择性搜索可生成 22,491 个区域，速度分别是 [4, 9] 的 25 倍和 13 倍，而且得分最高，达到 0.730 MABO。

图 7 显示了每个类别区域的平均最佳重叠率。对于除自行车以外的所有类别，我们的选择性搜索始终具有相对较高的 ABO 分数。由于自行车是线框对象，因此很难准确划分，因此在区域定位而非对象定位时，自行车的性能低得不成比例。

如果将我们的方法与其他方法进行比较，[9]的方法在火车类中效果更好，而我们的 "质量 "方法在其他类中的得分相近或更好。对于鸟、船、公共汽车、

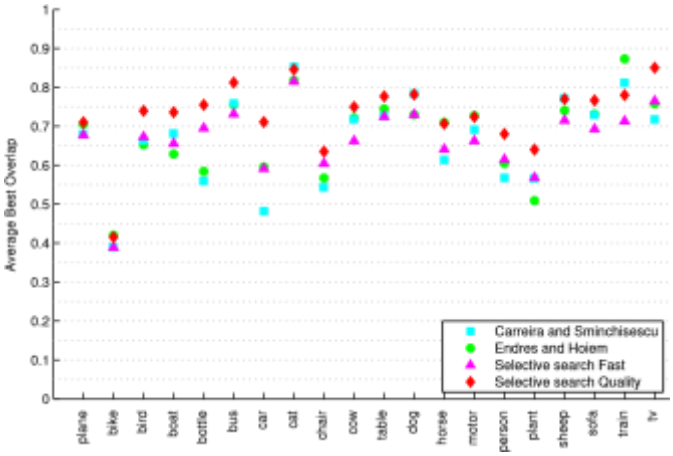

图 7：我们的方法与其他方法在 Pascal 2007 测试集上每个类别的平均最佳重叠得分比较。除 train 外，我们的 "质量 "方法始终能获得更好的平均最佳重叠分数。

椅子、人、植物和电视，我们的 "质量 "方法的 ABO 值要高出 0.05。对于汽车，我们的 ABO 值提高了 0.12，而对于瓶子，我们的 ABO 值甚至提高了 0.17。观察表 6 中 ABO 分数的变化，我们发现选择性搜索的变化略低于其他方法："质量 "的 MABO 为 0.093，[9] 为 0.108。不过，由于有线框自行车，这个分数是有偏差的：如果没有自行车，差异会更加明显。质量 "选择性搜索的标准偏差为 0.058，[9] 的标准偏差为 0.100。这再次表明，依靠多种互补策略而不是单一策略能产生更稳定的结果。

图 8 显示了我们的方法和 [4, 9] 的几个分割实例。在第一幅图像中，其他方法很难将瓶子的白色标签和书分开。在我们的案例中，我们的一种策略忽略了颜色，而 "填充 "相似性（公式 5）则有助于将瓶子和标签组合在一起。缺失的瓶子部分布满灰尘，在瓶子部分形成之前已经与桌子合并，因此 "填充 "在这里没有帮助。第二张图像是暗色图像的一个例子，由于使用了多种色彩空间，我们的算法在该图像上取得了很好的效果。在这张特殊图像中，部分强度不变的 Lab 色彩空间有助于隔离汽车。由于我们没有使用 [3] 中的轮廓检测方法，我们的方法有时会生成边界不规则的片段，第三幅猫的图像就说明了这一点。最后一张图片展示了一个非常困难的例子，只有 [4] 可以提供准确的分割。

现在，由于选择性搜索的性质，与其让这些方法相互对立，不如看看它们如何互补更有意思。由于[4, 9]的算法截然不同，根据我们的多样化假设，两者的结合应该是有效的。事实上，从表 6 的下半部分可以看出，与我们的 "快速 "选择性搜索相结合，

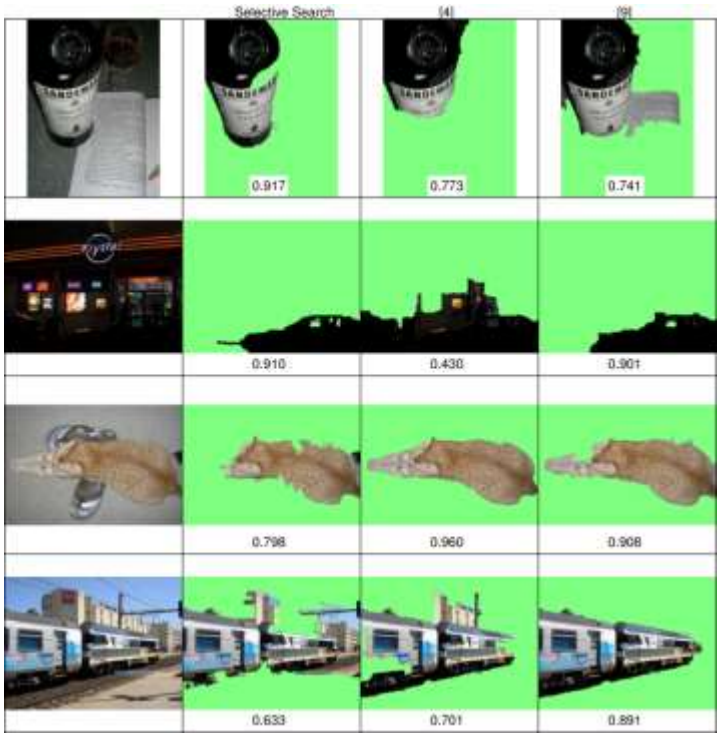

图 8：选择性搜索、[4] 和 [9] 的定性比较。对于我们的方法，我们观察到：忽略颜色可以找到瓶子，多种颜色空间有助于暗色图像（汽车），不使用 [3] 有时会导致不规则边界，如猫。

可以在 6438 个地点获得 0.737 MABO。与我们的 "高质量 "选择性搜索相比，用更少的位置得到了更高的 MABO。将[4, 9]与我们的 "高质量 "采样相结合，可在 25,355 个地点获得 0.758 MABO。这是一个很好的提高，只增加了少量地点。

总之，选择性搜索对于根据区域生成物体位置非常有效。多种策略的使用使其对各种图像条件和物体类别都具有鲁棒性。将 [4]、[9] 和我们的分组算法结合到单一的选择性搜索中，显示出了很好的改进效果。考虑到这些改进，并考虑到在选择性搜索中还可以使用更多不同的分区算法，我们很有兴趣看看我们的选择性搜索范例在计算效率、物体位置数量和物体位置质量方面还能走多远。

5.3 物体识别

在本节中，我们将使用 Pascal VOC 2010 检测任务对我们的选择性搜索策略进行评估。

我们的选择性搜索策略可以使用昂贵而强大的图像表征和机器学习技术。在本节中，我们将在第 4 节所述的基于词袋的物体识别框架内使用选择性搜索。与穷举式搜索相比，选择性搜索减少了目标

Participant	Flat error	Hierarchical error
University of Amsterdam (ours)	0.425	0.285
ISI lab., University of Tokyo	0.565	0.410

表 8：2011 年 ImageNet 大规模视觉识别挑战赛 (ILSVRC2011) 的结果。如果根据 WordNet 层次结构预测的类别在语义上与真实类别相似，则层次错误对错误的惩罚较轻。

位置的数量，因此使用这种强大的词袋实现是可行的。

为了说明计算要求：按像素提取三个 SIFT 变体加上视觉单词分配大约需要 10 秒钟，每幅图像只需完成一次。最后一轮 SVM 学习在 GPU 上进行约 30,000 个训练实例[33]，每个类别耗时约 8 小时，这是在 Pascal VOC 2010 上进行两轮底片挖掘后的结果。硬底片的挖掘是并行进行的，在 10 台机器上进行一轮挖掘大约需要 11 个小时，即每张图像大约需要 40 秒。其中 30 秒用于计算视觉词频，每类 0.5 秒用于分类。测试需要 40 秒来提取特征、分配视觉单词和计算视觉单词频率，之后每个类别需要 0.5 秒来进行分类。相比之下，[12]的代码（没有级联，就像我们的版本一样）每幅图像每类的测试时间略少于 4 秒。对于 20 个 Pascal 类别来说，这使得我们的框架在测试过程中速度更快。

我们使用官方评估服务器对结果进行评估。由于测试数据尚未公布，因此本次评估是独立的。我们与竞赛的前四名进行了比较。需要注意的是，前四名中的所有方法都是基于穷举搜索的，使用的是基于 [12] 的部分模型和 HOG 特征的变体，而我们的方法则通过使用选择性搜索和词袋特征有很大不同。结果如表 7 所示。

结果表明，我们的方法对平面、猫、牛、桌子、狗、植物、羊、沙发和电视等类的结果最好。除了桌子、沙发和电视，这些类别都是非刚性的。这是意料之中的，因为从理论上讲，词袋比 HOG 特征更适合这些类别。事实上，对于自行车、瓶子、公共汽车、汽车、人和火车这些刚性类别，基于 HOG 的方法表现更好。但刚性类别 "tv "是个例外。这可能是因为我们的选择性搜索在定位 tv 方面表现出色，见图 6。

在 Pascal 2011 挑战赛中，有几种方法的得分明显高于我们的方法。这些方法使用词袋作为其基于部分的模型所发现位置的附加信息，从而获得了更高的检测精度。但有趣的是，通过使用词袋来检测

位置，我们的方法在许多类别中获得了更高的总召回率[10]。

最后，我们的选择性搜索使我们得以参加 2011 年 ImageNet 大规模视觉识别挑战赛（ILSVRC2011）的检测任务，如表 8 所示。该数据集包含 1,229,413 幅训练图像和 100,000 幅测试图像，共有 1,000 种不同的物体类别。由于从选择性搜索位置提取的特征可重复用于所有类别，因此测试可以加速。例如，使用 [30] 的快速词袋框架，提取 SIFT 描述符和两种颜色变体的时间为 6.7 秒，分配到视觉词的时间为 1.7 秒。使用 1x1、2x2 和 3x3 空间金字塔划分法提取全部 172,032 维特征需要 14 秒。然后在金字塔层级上进行级联分类，每类需要 0.3 秒。对于 1,000 个类别，每幅图像的测试总过程需要 323 秒。相比之下，使用 [12] 基于部分的框架，每幅图像的每个类别需要 3.9 秒，因此每幅图像的测试时间为 3900 秒。这清楚地表明，位置数量的减少有助于扩展到更多的类别。

我们的结论是，与穷举搜索相比，选择性搜索可以使用更昂贵的特征和分类器，并且随着类别数量的增加，其扩展性更好。

5.4 Pascal VOC 2012

由于 2012 年 Pacal VOC 是最新的 VOC 数据集，也可能是最终的 VOC 数据集，因此我们简要介绍了该数据集的结果，以便将来与我们的工作进行比较。我们介绍了使用 TRAIN+VAL 集的方框质量、TRAIN+VAL 分割部分的分段质量，以及我们使用官方评估服务器在 TEST 集上使用 1x1、2x2、3x3 和 4x 4 空间金字塔的定位框架。

定位质量结果见表 9。我们看到，盒式定位的结果略高于 Pascal VOC 2007。然而，在分段定位方面，结果则要差一些。这主要是因为 2012 年的分段集难度要大得多。

2012 年检测挑战的平均精度为 0.350。这与 2010 年在 Pascal VOC 上获得的 0.351 MAP 相似。

5.5 定位质量的上限

在本实验中，我们研究了选择性搜索位置与词袋特征识别准确率最佳位置的接近程度。我们在 Pascal VOC 2007 测试集上进行了这项实验。

Boxes TRAIN+VAL 2012	MABO	# locations
"Fast"	0.814	2006
"Quality"	0.886	10681
Segments TRAIN+VAL 2012	**MABO**	**# locations**
"Fast"	0.512	3482
"Quality"	0.559	22073

表 9：Pascal VOC 2012 TRAIN+VAL 上的位置质量。

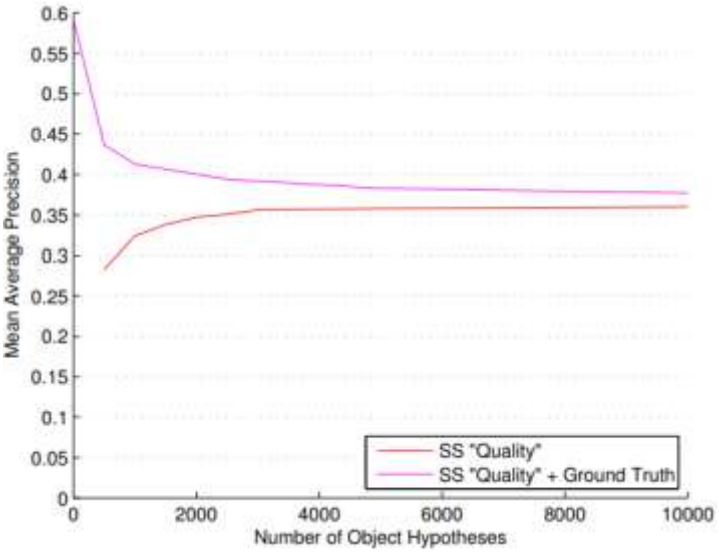

图 9：物体识别框架中方框选择的理论上限。红色曲线表示使用 "高质量 "选择性搜索方法的前 n 个位置的性能，500 个位置的 MABO 为 0.758，3000 个位置的 MABO 为 0.855，10,000 个位置的 MABO 为 0.883。洋红色曲线表示使用相同的前 n 个位置的性能，但现在结合了地面实况，即位置质量的上限（MABO = 1）。在 10,000 个位置时，我们的目标假设集接近最佳目标识别准确率。

图 9 中的红线显示了在使用 "高质量 "选择性搜索方法的前 n 个方框时，我们的物体识别系统的 MAP 分数。使用前 500 个对象位置时，MAP 为 0.283，MABO 为 0.758。使用前 3000 个对象位置时，性能迅速上升到 0.356 MAP，MABO 为 0.855；使用全部 10,097 个对象位置时，性能达到 0.360 MAP，MABO 为 0.883。

洋红色线条显示的是我们的物体识别系统的性能，如果我们将地面实况的物体位置包含到假设集中，则代表了 "完美 "质量的物体假设集，其 MABO 分数为 1。当仅使用地面实况框时，MAP 为 0.592，这是我们物体识别系统的上限。然而，在每幅图像仅使用 500 个位置的情况下，这一得分迅速下降到 0.437 MAP。值得注意的是，当使用全部 10,079 个方框时，性能下降到 0.377 MAP，仅比不包括地面实况时多 0.017 MAP。这表明，在 10,000 个对象位置时，我们的假设集接近于我们的识别框

System	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv
NLPR	.533	**.553**	**.192**	**.210**	.300	.544	.467	.412	**.200**	.315	.207	.303	.486	.553	.465	.102	.344	.265	**.503**	.403
MIT UCLA [38]	.542	.485	.157	.192	.292	**.555**	.435	.417	.169	.285	.267	.309	.483	.550	.417	.097	.358	.308	.472	.408
NUS	.491	.524	.178	.120	.306	.535	.328	.373	.177	.306	.277	.295	**.519**	**.563**	.442	.096	.148	.279	.495	.384
UoCTTI [12]	.524	.543	.130	.156	**.351**	.542	**.491**	.318	.155	.262	.135	.215	.454	.516	**.475**	.091	.351	.194	.466	.380
This paper	**.562**	.424	.153	.126	.218	.493	.368	**.461**	.129	**.321**	**.300**	**.365**	.435	.529	.329	**.153**	**.411**	**.318**	.470	**.448**

表 7：Pascal VOC 2010 检测任务测试集的结果。我们的方法是唯一基于词袋的物体识别系统。它在 9 个主要为非刚性物体的类别中得分最高，两者之间的差距高达 0.056 AP。其他方法基于基于部分的 HOG 特征，在大多数刚性物体类别中表现较好。

架所能达到的最佳值。最有可能的解释是我们使用了 SIFT，而 SIFT 被设计为具有移位不变性[21]。这使得图 5 所示的近似方框质量仍然足够好。然而，由 10,000 个方框组成的 "完美 "对象假设集与我们的差距很小，这表明我们已经达到了词袋不变性程度可能会产生不利影响而非有利影响的程度。

随着方框数量的增加，"完美 "假设集的减少是由于问题难度的增加：更多的方框意味着更高的可变性，这使得物体识别问题变得更加困难。之前我们假设，穷举式搜索会检查图像中所有可能的位置，这使得物体识别问题变得困难。为了测试选择性搜索是否能缓解这一问题，我们还在穷举搜索中使用了我们的词袋物体识别系统，并使用了 [12] 中的位置。结果 MAP 为 0.336，MABO 为 0.829，每类对象位置数为 100,000 个。使用 2,000 个位置并进行选择性搜索，也能获得相同的 MABO。在 2,000 个位置时，目标识别准确率为 0.347。这表明，与穷举搜索相比，选择性搜索通过减少位置的可能变化，确实使问题变得更容易解决。

总之，物体假设的质量和数量与物体识别的准确性之间存在权衡。首先，高质量的物体位置是识别物体的必要条件。在不牺牲质量的前提下，减少对象假设的采样数量可以使分类问题变得更容易，并有助于提高分类结果。值得注意的是，在合理的 10,000 个位置上，我们的对象假设集接近于词袋识别系统的最佳值。这表明，我们的位置质量很高，现在需要的特征具有比通常的词袋识别更高的判别能力。

6 结论

本文建议将分割技术用于选择性搜索。我们发现，图像本身具有层次性，一个区域形成一个物体的原因多种多样。因此，单一的自下而上分组算法永远无法捕捉到所有可能的物体位置。为了解决这个问题，我们引入了选择性搜索，其主要原理是使用一系列不同的互补和分层分组策略。这使得选择性搜索变得稳定、鲁棒性强，并且与对象类别无关，对象类型从刚性（如汽车）到非刚性（如猫），理论上也包括非定形（如水）。

在对象窗口方面，结果显示我们的算法优于[2]的 "对象性"，我们的快速选择性搜索在 2,134 个位置上达到了 0.804 平均最佳重叠率的质量。与文献[4, 9]相比，我们的算法在生成窗口的质量和数量之间进行了类似的权衡，在 790 个位置（这是他们生成的最大值）时，平均最佳重叠率约为 0.790。然而，我们的算法要快 13-59 倍。此外，它还能在每幅图像上创建多达 10,097 个位置，产生高达 0.879 的 MABO。

就对象区域而言，将我们的算法与 [4, 9] 结合使用，质量有了显著提高（MABO 从 0.730 提高到 0.758），这表明采用我们的多样化范式仍有改进的余地。

最后，我们证明了选择性搜索可成功用于创建一个良好的基于词袋的定位和识别系统。事实上，我们的研究表明，对于我们的基于词袋的物体识别系统来说，选择性搜索位置的质量接近最佳。

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 2, 6

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 2012. 3, 8, 10, 13

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour´ detection and hierarchical image segmentation. *TPAMI*, 2011. 1, 2, 3, 4, 8, 10, 11

[4] J. Carreira and C. Sminchisescu. Constrained parametric mincuts for automatic object segmentation. In *CVPR*, 2010. 2, 3, 8, 9, 10, 11, 13

[5] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, 2007. 3

[6] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *TPAMI*, 24:603–619, 2002. 1, 3

[7] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Statistical Learning in Computer Vision*, 2004. 5

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1, 2, 3, 5

[9] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010. 2, 3, 6, 8, 9, 10, 11, 13

[10] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. Overview and results of the detection challenge. The Pascal Visual Object Classes Challenge Workshop, 2011. 12

[11] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2010. 6

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 32:1627–1645, 2010. 1, 2, 3, 5, 6, 8, 11, 12, 13

[13] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient GraphBased Image Segmentation. *IJCV*, 59:167–181, 2004. 1, 3, 4, 5, 7

[14] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *TPAMI*, 23:1338–1350, 2001. 4

[15] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik. Recognition using´ regions. In *CVPR*, 2009. 2

[16] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009. 1, 2, 3, 5, 6, 8

[17] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *TPAMI*, 31:2129–2142, 2009. 2, 5

[18] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 5

[19] F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *CVPR*, 2010. 2

[20] C. Liu, L. Sharan, E.H. Adelson, and R. Rosenholtz. Exploring features in a bayesian framework for material recognition. In *Computer Vision and Pattern Recognition 2010*. IEEE, 2010. 4

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 5, 13

[22] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008. 5

[23] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *CVPR*, 2009. 3

[24] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002. 7

[25] Florent Perronnin, Jorge Sanchez, and Thomas Mensink. Im-´ proving the Fisher Kernel for Large-Scale Image Classification. In *ECCV*, 2010. 5

[26] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22:888–905, 2000. 1

[27] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 5

[28] Soeren Sonnenburg, Gunnar Raetsch, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio de Bona, Alexander Binder, Christian Gehl, and Vojtech Franc. The shogun machine learning toolbox. *JMLR*, 11:1799–1802, 2010. 5

[29] Z. Tu, X. Chen, A. L. Yuille, and S. Zhu. Image parsing: Unifying segmentation, detection and

recognition. *International Journal of Computer Vision*, Marr Prize Issue, 2005. 1

[30] J.R.R. Uijlings, A.W.M. Smeulders, and R.J.H. Scha. Realtime visual concept classification. *IEEE Transactions on Multimedia*, In press, 2010. 5, 12

[31] K. E. A. van de Sande and T. Gevers. Illumination-invariant descriptors for discriminative visual object categorization. Technical report, University of Amsterdam, 2012. 5

[32] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek.
Evaluating color descriptors for object and scene recognition. *TPAMI*, 32:1582–1596, 2010. 5, 12

[33] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Empowering visual categorization with the GPU. *TMM*, 13(1):60–70, 2011. 11

[34] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 3, 5, 6, 8

[35] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, volume 1, pages 511– 518, 2001. 1

[36] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57:137–154, 2004. 2, 3

[37] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV*, 2010. 5

[38] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010. 13