**Literature Survey文献综述**

**回顾文献中的相关技术，以及理解你所选择的技术的任何必要背景。**

**Method 方法**

**证明并解释你所实施的技术的选择，必要时使用相关参考资料和理论。**

**》〉Task1**

**\*\*\*\*hog+SVM**

The target detection task involves determining the presence of a target in an image and its location. HOG is a histogram of gradient-based feature descriptors, which is often used in face recognition to compute and count histograms of gradient directions in localized areas of an image to form features.

When computing the HOG Descriptor, there will be the following parameters. windowSize, blockSize, blockStep, cellSize, binNum. Each window will be divided into blocks, and each block will be divided into cells. nbins represent the number of directions of a statistical gradient in a cell. A histogram of the gradient of each pixel in the cell unit is captured. Finally, these histograms are combined to form a feature descriptor.

The number of cells in a block is:

A=(blockSize.width)/(cellSize.width)\*(blockSize.height)/(cellSize.height)

Each block contains binNum\*A gradient histogram.

The number of blocks in a window is:

B=((windowSize.width-blockSize.width)/(blockStep.width)+1)\*((windowSize.height-blockSize.height)/(blockStep.height)+1)

One window contains 9AB gradient histograms.

There are two other issues that need to be solved when using hog: scale and position. The scale problem can be solved by resizing the image. To solve the position problem, image pyramids and sliding windows are used.

\*\* Image Pyramid

An image pyramid is a multi-scale representation of an image, which helps solve the problem of target detection at different scales. The more layers of the pyramid, the slower the computation will be, but the more accurate the results will be to some extent.

\*\*Sliding window

Setting the window size to slide in the image, sliding window detection and using the image pyramid to detect each part, this is to detect objects at multiple scales. The sliding window solves the localization problem by scanning smaller areas of a larger image and thus repeating the scan at different scales of the same image. However, there is also the problem of region overlap, which requires non-maximal suppression to solve.

\*\*non\_max\_suppression

Non-maximal suppression is widely used in calculator vision tasks. Since in target detection, multiple candidate boxes are often generated for the same target, and they overlap. It is necessary to use non-maximal suppression to find the best target candidate frame and remove the redundant candidate frames.

After HOG, a classifier is needed to classify the features. Generally, support vector machine (SVM) will be chosen.

In machine learning, a support vector machine (SVM) is a supervised learning model and associated learning algorithm for analyzing data in classification and regression analysis. Given a set of training instances, each of which is labeled as belonging to one or the other of two categories, the SVM training algorithm creates a model that assigns new instances to one of the two categories, making it a non-probabilistic binary linear classifier.

**\*\*\*\*根据颜色阈值来区分（原理eddie）**

**Experiment Setup**

**解释实验设置和*评价方法*。**

**\*\*HOG+SVM**

1. Read the data and obtain positive and negative samples, positive samples are obtained by csv cropping, negative samples are obtained by randomly taking the areas of the image other than plants.
2. Resize all the images. This step is to speed up the operation and solve the size problem in hog. While resizing the image, you should also resize the data in the csv so that the image and the csv data can be matched. The size of Resize is set to (1647, 1158) by comparing the size of the existing images.
3. Calculate the hog characteristics of the positive and negative samples and add a corresponding label for all samples. The label for positive samples is 1 and the label for negative samples is -1.
   1. hog = cv2.HOGDescriptor(winSize, blockSize, blockStep, cellSize, binNum)

winSize = (128, 128)

blockSize = (16, 16)

blockStep = (8, 8)

cellSize = (8, 8)

binNum = 9

Depending on the parameters, one window will produce 8100 gradient histograms.

1. Create the SVM model and perform the first training.
2. Use hardexample to optimize the model based on the results of the first training. It uses all the detected rectangular boxes from the first training of the classifier when performing plant detection on the negative sample original image. These regions are false, and saving these false regions as images, adding them to the initial negative sample set, and re-training the SVM can significantly reduce false predictions.
3. Extract the hog features of the hardexample and add them to the previous sample features. Add the label corresponding to the hardexample to the label\_list (-1). Retrain all hog features and save the model.
4. Detection and identification. Need to locate the plant and therefore need to get the area information. So use hog.setSVMDetector and hog.detectMultiScale.
5. Non-maximal suppression optimizes the overlapping detection regions to obtain the final predicted bbox.
6. Average Precision is calculated by comparing the obtained predicted bbox with the ground truth in the csv file.

**\*\*颜色阈值**

1. Read the input data. 将绿色提取出来，进行预处理**（eddie）**
2. The image is black in the background and white in the foreground. The binary image is obtained by reversing the color so that the background is 255 white and the foreground is 0 black.
3. Use the two-pass connected components algorithm. In the first pass, assign temporary labels to each pixel and store the labels as equivalent to their neighbors' labels. The smallest label of the equivalents is chosen to replace the temporary label in the second pass. A rough prediction bbox is obtained.
4. Due to the imperfection of the binary diagram, some of the stems and leaves of the plant are missing. A plant may have been divided into several parts. Therefore, these parts need to be merged. The label is redistributed by roughly predicting the bbox.
   1. Create a new label\_image with all 0. Find the xmin, ymin, xmax, ymax of each label to get a rectangular area. Assign a label to each rectangular region. bboxes are arranged in descending order according to the area of the region.
   2. If the region's label is all 0, then a new label is assigned to the region. If the region's label is not all 0, but the region area is greater than the threshold, assign a new label to the region as well. When the area of the region is less than the threshold and the label is not all 0, then find the value in the label that is not 0 and assign the region to this value.
      1. Select the first 5% of the descending order of the AREA in the rough bbox as the threshold value.
      2. If the area of the region is larger than the threshold, infer that the rectangular box is likely to be a plant.
5. A new predicted bbox is obtained. AP is calculated by comparing the obtained bbox with the ground truth.

**\*\*AP**

One of the common evaluation metrics used in target detection problems is Average Precision. AP actually calculates the relationship between ground truth (true labels) and the prediction box. The calculation of AP also needs precision and recall.

In general, the calculation of Average Precision and Recall requires True Positives, False Positives, True Negatives and False Negatives.

To get True Positives and False Positives, calculate IoU (Intersection over Union).

\*\*\*IoU

IoU is one of the metrics for evaluating the correctness of bounding boxes. IoU calculates the ratio of the intersection and concatenation of the prediction box and ground truth. For each class, the area where the prediction box and ground truth overlap is the intersection, and the total area across is the union.

When calculating the IoU, a threshold is used, the most common threshold is 0.5. If the IoU is greater than the threshold, then the prediction is considered True Positive. if the IoU is less than the threshold, then the prediction is considered False Positive.

True Negatives are difficult to calculate in target detection because any part of the image that is not a predicted object is considered Negative. therefore, only objects that are not detected in the model detection are calculated as False Negatives.

Steps.

1. Calculate the IoU of each predicted box in the bbox with ground truth and take the maximum value of the IoU and compare it with the IoU threshold to determine the number of TP and FP.
2. Calculate accuracy and recall rates.
3. 11 different values of recall were selected according to the references [0, 0.1, ... , 0.9, 1.0]) and calculate the average of the accuracy of the 11 recalls as the value of AP.

**Results**

**提供统计和视觉结果，同时讨论方法性能和实验结果。**

Input data: Ara2012:Tray/Ara2012/\*\_rgb.png(16files), Tray/Ara2012/\*\_bbox.csv (16 files)

The two methods of target detection for 16 images in the Ara2012 file yielded the AP results shown in the table below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Image | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Hog+svm | 0.2545 | 0.4545 | 0.3409 | 0.6294 | 0.5289 | 0.6234 | 0.8182 | 0.8182 | 0.8182 | 1 | 0.9091 | 0.9091 | 0.8182 | 1 | 0.9091 | 0.9091 |
| Colorthreshold | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

Time cost(seconds)

|  |  |
| --- | --- |
| Hog+svm | 19.797090 |
| Colorthreshold | 410.927392 |

图片效果展示

Image1

|  |  |  |
| --- | --- | --- |
| 图形用户界面, 应用程序  描述已自动生成 | 图形用户界面, 应用程序  描述已自动生成 | 图形用户界面, 应用程序  描述已自动生成 |
| Ground\_truth | HOG+SVM(AP:0.2545) | Colorthreshold(AP:1.0) |

Image8

|  |  |  |
| --- | --- | --- |
| 图形用户界面, 应用程序  描述已自动生成 | 图片包含 夹, 行, 游戏机, 一群  描述已自动生成 | 图形用户界面, 应用程序  描述已自动生成 |
| Ground\_truth | HOG+SVM(AP:0.8182) | colorthreshold(AP:1.0) |

Image10

|  |  |  |
| --- | --- | --- |
| 图形用户界面, 应用程序  描述已自动生成 | 图形用户界面  描述已自动生成 | 图形用户界面, 应用程序  描述已自动生成 |
| Ground\_truth | HOG+SVM(AP:1.0) | colorthreshold(AP:1.0) |

**Discussion and conclusions**

**\*\*HOG+SVM**

1. 非极大值抑制的阈值选择，需要进行一些调整。在这里，选用了threshold=0.06。
2. 样本的多少，负样本的选取，样本很重要，要贴合项目实际环境去获取图片数据集。之前有尝试在读取图片数据后，随机提取与正样本数量相同的负样本。问题在于：选取的不够全面，仍会导致最后的检测不准确。所以最后是直接随机截取样本保存后，再进行人工筛选。
3. 不同的图片的效果不同。尤其是当植物较小的时候，检测效果不好。考虑是否在正样本的数据中多加入一些不同的较小的植物图片。
4. 运行时间较短，如果多次运行，可以直接调用保存的模型。

**\*\*颜色阈值**

1. 在预处理部分，如果过多的去噪音，那么就会导致植物的茎叶不明显。同一株植物叶子与叶子之间出现断连。
2. 在得到第一次粗略的预测bbox时，要剔除一些label的个数很少的点。实际上可以看作是手动去噪音。
3. 如果两株植物靠的太近或者两株植物的叶片发生了重叠，那么就会被认为是同一株植物。虽然可以定位植物的位置，但是会影响植物个数的判断。
4. 使用传统方法，因此在运行过程中，对图片的像素点进行了多次遍历。时间复杂度大，运行时间较长。

**\*\*两个方法进行总结比较**

--》不同点

1. Hog+svm需要大量的正负样本，并且样本的选取尽量需要贴合项目实际环境去获取。
2. 两者本质的不同，hog+svm是机器学习，颜色阈值就是传统方法。
3. Hog+svm适用于同类型的目标进行检测，例如行人检测。颜色阈值适用于单色物体的目标检测与定位。
4. Hog+svm 用时短，颜色用时长。但是前者对于检测较小的植物效果很差，检测较大的植物效果好。而颜色阈值的方法受到植物大小的影响较小。
5. Hog+svm的框都是固定的尺寸比例，不会随着植物的大小进行改变。因此预测的框与植物ground\_truth框的偏差较大。使用颜色阈值的方法所画的框会随着植物的形状有所改变。