JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Object Detection and Counting using Image Segmentation Techniques

***Yethiraj.N.G.**

Assistant Professor of Computer Science, Maharani's Science College for Women, Bengaluru.

****Dr.B.Basavaprasad,**

Assistant Professor of Computer Science, Govt. First Grade College, Raichur.

Abstract

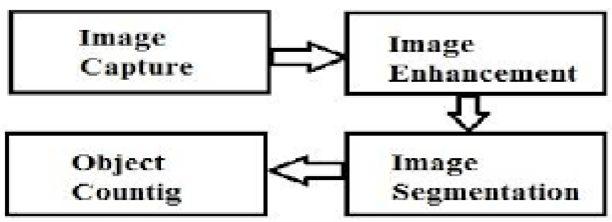
This paper attempts to study significance of **mage segmentation** ; the process of partitioning a digital image into multiple **image segments**, also known as **image regions** or **image objects** (sets of pixels)/ counting. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property , such as color, intensity, or texture. Adjacent regions are significantly different color respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like marching cubes. Instance division are tough since this technology needs accurate perception of all the objects present in a picture while also accurately segments every occurrence. For that reason merges essentials by the traditional computer vision responsibilities of physical object sensing, the purpose is to knowing the categorise each physical entity's and localise to each one by means of bounding box. Then next is to use the semantic sectionalization to categorise every picture element into a assured equipoise collection and also perfoms lack off differentiating physical object occurrence. Certain this might be a difficult process to reach good outcomes. But, we illustrate that an unexpectedly straightforward, flexible, and high-speed system that can exceed previous state-ofthe-art occurrence segmentation outcome

Keywords: counter; **object counting**; **object detection**; deep learning; segmentation; computer vision

Introduction

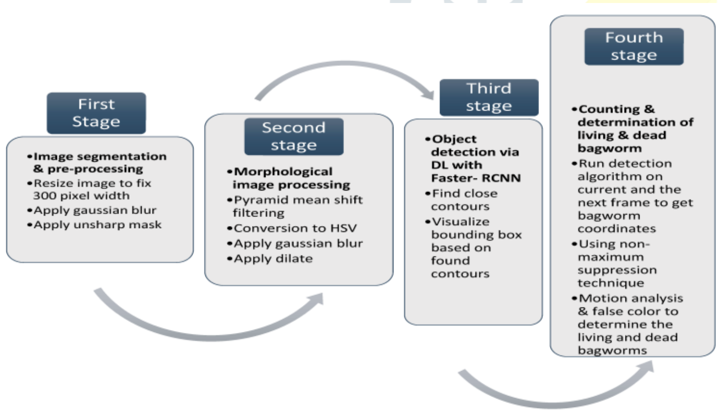
Object Detection is by far one of the most important fields of research in Computer Vision. Researchers have for a long time been interested in this field, but significant results were produced in the recent years owing to the rise of Convnets as feature extractors and Transfer Learning as method of passing on previous knowledge. Early object detectors were based on handcrafted features, and employed a sliding window based approach which was computationally inefficient and less accurate. Modern techniques include Region Proposal Methods, Single Shot Methods, Anchor Free Methods and so on . The simplest method of image segmentation is called the thresholding method. This method is based on a clip-level (or a threshold value) to turn a gray-scale image into a binary image.



ieneralized Framework for Object Counting using Image Processing

The key of this method is to select the threshold value (or values when multiple-levels are selected). Several popular methods are used in industry including the maximum entropy method, balanced histogram thresholding, Otsu's method (maximum variance), and k-means clustering.

Recently, methods have been developed for thresholding computed tomography (CT) images. The key idea is that, unlike Otsu's method, the thresholds are derived from the radiographs instead of the (reconstructed) image.



New methods suggested the usage of multi-dimensional fuzzy rule-based non-linear thresholds. In these works decision over each pixel's membership to a segment is based on multi-dimensional rules derived from fuzzy logic and evolutionary algorithms based on image lighting environment and application.

Objective:

This paper intends to explore and analyze; **Counting objects with computer vision** and apply machine learning methods to recognize and count the instances with **Image Segmentation Techniques** in which particular objects appear within each frame

Object detection : classification and localization

Object detection combines classification and localization to determine what objects are in the image or video and specify where they are in the image. Object detection is useful in identifying objects in an image or video. Below, the image on the left illustrates classification, in which the classes *Donut* and *Coffee* are identified. The image on the right illustrates object detection by surrounding the members of each class — donut and coffee — with a bounding box.

Use cases for object detection include facial detection with any post-detection analysis; for example, expression detection, age estimation or drowsiness detection. Many real-time object detection applications exist for traffic management, such as vehicle detection systems based on traffic scenes.

As described above, the most popular approaches to computer vision are classification and object detection to identify objects present in an image and specify their position. But many use cases call for analyzing images at a lower level than that. That is where image segmentation comes in. Any image consists of both useful and useless information, depending on the user's interest. Image segmentation separates an image into regions, each with its particular shape and border, delineating potentially meaningful areas for further processing, like classification and object detection. The regions may not take up the entire image, but the goal of image segmentation is to highlight foreground elements and make it easier to evaluate them. Image segmentation provides pixel-by-pixel details of an object, making it different from classification and object detection.

Connectivity-based clustering

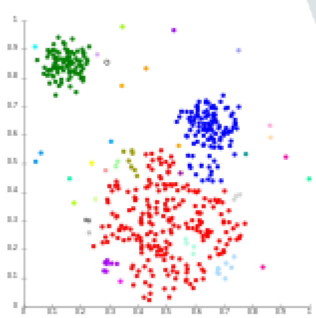
Connectivity-based clustering, also known as *hierarchical clustering*, is based on the core idea of objects being more related to nearby objects than to objects farther away. These algorithms connect "objects" to form "clusters" based on their distance. A cluster can be described largely by the maximum distance needed to connect parts of the cluster. At different distances, different clusters will form, which can be represented using a dendrogram, which explains where the common name "hierarchical clustering" comes from: these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances. In a dendrogram, the y-axis marks the distance at which the clusters merge, while the objects are placed along the x-axis such that the clusters don't mix.

Connectivity-based clustering is a whole family of methods that differ by the way distances are computed. Apart from the usual choice of distance functions, the user also needs to decide on the linkage criterion (since a cluster consists of multiple objects, there are multiple candidates to compute the distance) to use. Popular choices are known as single-linkage clustering (the minimum of object distances), complete linkage clustering (the maximum of object distances), and UPGMA or WPGMA ("Unweighted or Weighted Pair Group Method with Arithmetic Mean", also known as average linkage clustering). Furthermore, hierarchical clustering can be agglomerative (starting with single elements and aggregating them into clusters) or divisive (starting with the complete data set and dividing it into partitions).

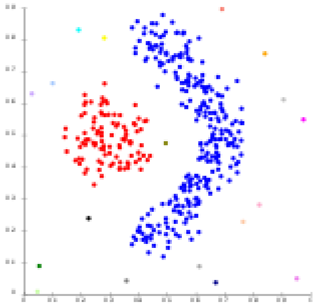
Image segmentation techniques use different algorithms.

Algorithm	Description
Edge Detection Segmentation	Makes use of discontinuous local features of an image to detect edges and hence define a boundary of the object.
Mask R-CNN	Gives three outputs for each object in the image: its class, bounding box coordinates, and object mask
Segmentation based on Clustering	Divides the pixels of the image into homogeneous clusters.
Region-Based Segmentation	Separates the objects into different regions based on threshold value(s).

Linkage clustering examples



Single-linkage on Gaussian data. At 35 clusters, the biggest cluster starts fragmenting into smaller parts, while before it was still connected to the second largest due to the single-link effect.



Single-linkage on density-based clusters. 20 clusters extracted, most of which contain single elements, since linkage clustering does not have a notion of "noise".

The optimization problem itself is known to be NP-hard, and thus the common approach is to search only for approximate solutions. A particularly well known approximate method is Lloyd's algorithm, often just referred to as "*k-means algorithm*" (although another algorithm introduced this name). It does however only find a local optimum, and is commonly run multiple times with different random initializations. Variations of *k*-means often include such optimizations as choosing the best of multiple runs, but also restricting the centroids to members of the data set (*k*-medoids), choosing medians (*k*-medians clustering), choosing the initial centers less randomly (*k*-means++) or allowing a fuzzy cluster assignment (fuzzy c-means).

Most *k*-means-type algorithms require the number of clusters – *k* – to be specified in advance, which is considered to be one of the biggest drawbacks of these algorithms. Furthermore, the algorithms prefer clusters of approximately similar size, as they will always assign an object to the nearest centroid. This often leads to incorrectly cut borders of clusters (which is not surprising since the algorithm optimizes cluster centers, not cluster borders).

```
from
skimage.morphology
import watershed
from scipy import ndimage

D = ndimage.distance_transform_edt(thr)
localMax = peak_local_max(D, indices=False, min_distance=40, labels=thr)
markers = ndimage.label(localMax, structure=np.ones((3, 3)))
labels = watershed(-D, markers, mask=thr)
ws = len(np.unique(labels)) -1
```

Compression based methods

Compression based methods postulate that the optimal segmentation is the one that minimizes, over all possible segmentations, the coding length of the data. The connection between these two concepts is that segmentation tries to find patterns in an image and any regularity in the image can be used to compress it. The method describes each segment by its texture and boundary shape. Each of these components is modeled by a probability distribution function and its coding length is computed as follows:

1. The boundary encoding leverages the fact that regions in natural images tend to have a smooth contour. This prior is used by Huffman coding to encode the difference chain code of the contours in an image. Thus, the smoother a boundary is, the shorter coding length it attains.

2. Texture is encoded by lossy compression in a way similar to minimum description length (MDL) principle, but here the length of the data given the model is approximated by the number of samples times the entropy of the model. The texture in each region is modeled by a multivariate normal distribution whose entropy has a closed form expression. An interesting property of this model is that the estimated entropy bounds the true entropy of the data from above. This is because among all distributions with a given mean and covariance, normal distribution has the largest entropy. Thus, the true coding length cannot be more than what the algorithm tries to minimize.

For any given segmentation of an image, this scheme yields the number of bits required to encode that image based on the given segmentation. Thus, among all possible segmentations of an image, the goal is to find the segmentation which produces the shortest coding length. This can be achieved by a simple agglomerative clustering method. The distortion in the lossy compression determines the coarseness of the segmentation and its optimal value may differ for each image. This parameter can be estimated heuristically from the contrast of textures in an image. For example, when the textures in an image are similar, such as in camouflage images, stronger sensitivity and thus lower quantization is required.

Image segmentation and primal sketch

There have been numerous research works in this area, out of which a few have now reached a state where they can be applied either with interactive manual intervention (usually with application to medical imaging) or fully automatically. The following is a brief overview of some of the main research ideas that current approaches are based upon.

The nesting structure that Witkin described is, however, specific for one-dimensional signals and does not trivially transfer to higher-dimensional images. Nevertheless, this general idea has inspired several other authors to investigate coarse-to-fine schemes for image segmentation. Koenderink proposed to study how iso-intensity contours evolve over scales and this approach was investigated in more detail by Lifshitz and Pizer. Unfortunately, however, the intensity of image features changes over scales, which implies that it is hard to trace coarse-scale image features to finer scales using iso-intensity information.

Lindeberg studied the problem of linking local extrema and saddle points over scales, and proposed an image representation called the scale-space primal sketch which makes explicit the relations between structures at different scales, and also makes explicit which image features are stable over large ranges of scale including locally appropriate scales for those. Bergholm proposed to detect edges at coarse scales in scale-space and then trace them back to finer scales with manual choice of both the coarse detection scale and the fine localization scale.

Gauch and Pizer studied the complementary problem of ridges and valleys at multiple scales and developed a tool for interactive image segmentation based on multi-scale watersheds. The use of multi-scale watershed with application to the gradient map has also been investigated by Olsen and Nielsen and been carried over to clinical use by Dam. Vincken et al. proposed a hyperstack for defining probabilistic relations between image structures at different scales. The use of stable image structures over scales has been furthered by Ahuja and his co-workers into a fully automated system. A fully automatic brain segmentation algorithm based on closely related ideas of multi-scale watersheds has been presented by Undeman and Lindeberg and been extensively tested in brain databases.

These ideas for multi-scale image segmentation by linking image structures over scales have also been picked up by Florack and Kuijper. Bijaoui and Rué associate structures detected in scale-space above a minimum noise threshold into an object tree which spans multiple scales and corresponds to a kind of feature in the original signal. Extracted features are accurately reconstructed using an iterative conjugate gradient matrix method.

Conclusion

Object counting is a not easy task in image processing. It is usually agreed in different areas of industries, research institutes, laboratories, agriculture industries among others. Accurately counting objects instances in a given picture or video frame is a difficult problem to solve in machine learning. Identifying the number of objects present in the image can be helpful for extra investigation in a spacious set of applications For high-dimensional data, many of the existing methods fail due to the curse of dimensionality, which renders particular distance functions problematic in high-dimensional spaces. This led to new clustering algorithms for high-dimensional data that focus on subspace clustering (where only some attributes are used, and cluster models include the relevant attributes for the cluster) and correlation clustering that also looks for arbitrary rotated ("correlated") subspace clusters that can be modeled by giving a correlation of their attributes. Examples for such clustering algorithms are CLIQUE and SUBCLU.

Ideas from density-based clustering methods (in particular the DBSCAN/OPTICS family of algorithms) have been adapted to subspace clustering (HiSC, hierarchical subspace clustering and DiSH) and correlation clustering (HiCO, hierarchical correlation clustering, 4C using "correlation connectivity" and ERiC exploring hierarchical density-based correlation clusters).

Several different clustering systems based on mutual information have been proposed. One is Marina Meilă's *variation of information* metric; another provides hierarchical clustering. Using genetic algorithms, a wide range of different fit-functions can be optimized, including mutual information. Also belief propagation, a recent development in computer science and statistical physics, has led to the creation of new types of clustering algorithms

References

1. Dasiopoulou, Stamatia, et al. "Knowledge-assisted semantic video object detection." IEEE Transactions on Circuits and Systems for Video Technology 15.10 (2005): 1210–1224.
2. Ling Guan; Yifeng He; Sun-Yuan Kung (1 March 2012). Multimedia Image and Video Processing. CRC Press. pp. 331–. ISBN 978-1-4398-3087-1.
3. Alsanabani, Ala; Ahmed, Mohammed; AL Smadi, Ahmad (2020). "Vehicle Counting Using Detecting-Tracking Combinations: A Comparative Analysis". 2020 the 4th International Conference on Video and Image Processing. pp. 48–54. doi:10.1145/3447450.3447458. ISBN 9781450389075. S2CID 233194604.
4. Wu, Jianxin, et al. "A scalable approach to activity recognition based on object use." 2007 IEEE 11th international conference on computer vision. IEEE, 2007.
5. Bochkovskiy, Alexey (2020). "Yolov4: Optimal Speed and Accuracy of Object Detection". arXiv:2004.10934 [cs.CV].

6. Dalal, Navneet (2005). "Histograms of oriented gradients for human detection" (PDF). Computer Vision and Pattern Recognition. 1.
7. Ross, Girshick (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation" (PDF). Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE. pp. 580–587. arXiv:1311.2524. doi:10.1109/CVPR.2014.81. ISBN 978-1-4799-5118-5. S2CID 215827080.
8. Girschick, Ross (2015). "Fast R-CNN" (PDF). Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448. arXiv:1504.08083. Bibcode:2015arXiv150408083G.
9. Shaoqing, Ren (2015). "Faster R-CNN". Advances in Neural Information Processing Systems. arXiv:1506.01497.
10. Pang, Jiangmiao; Chen, Kai; Shi, Jianping; Feng, Huajun; Ouyang, Wanli; Lin, Dahua (2019-04-04). "Libra R-CNN: Towards Balanced Learning for Object Detection". arXiv:1904.02701v1 [cs.CV].
11. Liu, Wei (October 2016). "SSD: Single shot multibox detector". Computer Vision – ECCV 2016. European Conference on Computer Vision. Lecture Notes in Computer Science. Vol. 9905. pp. 21–37. arXiv:1512.02325. doi:10.1007/978-3-319-46448-0_2. ISBN 978-3-319-46447-3. S2CID 2141740.
12. Redmon, Joseph (2016). "You only look once: Unified, real-time object detection". Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. arXiv:1506.02640. Bibcode:2015arXiv150602640R.
13. Redmon, Joseph (2017). "YOLO9000: better, faster, stronger". arXiv:1612.08242 [cs.CV].
14. Redmon, Joseph (2018). "Yolov3: An incremental improvement". arXiv:1804.02767 [cs.CV].
15. Wang, Chien-Yao (2021). "Scaled-YOLOv4: Scaling Cross Stage Partial Network". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). arXiv:2011.08036. Bibcode:2020arXiv201108036W.
16. Zhang, Shifeng (2018). "Single-Shot Refinement Neural Network for Object Detection". Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4203–4212. arXiv:1711.06897. Bibcode:2017arXiv171106897Z.
17. Lin, Tsung-Yi (2020). "Focal Loss for Dense Object Detection". IEEE Transactions on Pattern Analysis and Machine Intelligence. 42 (2): 318–327. arXiv:1708.02002. Bibcode:2017arXiv170802002L. doi:10.1109/TPAMI.2018.2858826. PMID 30040631. S2CID 47252984.
18. Zhu, Xizhou (2018). "Deformable ConvNets v2: More Deformable, Better Results". arXiv:1811.11168 [cs.CV].
19. Dai, Jifeng (2017). "Deformable Convolutional Networks". arXiv:1703.06211 [cs.CV].