Efficient Automatic Annotation of Binary Masks for Enhanced Training of Computer Vision Models

Dylan Seychell
Dept. of Artificial Intelligence
University of Malta
Msida, Malta
dylan.seychell@ieee.org

Matthew Kenely
Dept. of Artificial Intelligence
University of Malta
Msida, Malta
matthew.kenely@ieee.org

Matthias Bartolo
Dept. of Artificial Intelligence
University of Malta
Msida, Malta
matthias.bartolo@ieee.org

Carl James Debono
Dept. of Communications and Computer Engineering
University of Malta
Msida, Malta
c.debono@ieee.org

Mark Bugeja
Dept. of Artificial Intelligence
University of Malta
Msida, Malta
mark.bugeja@um.edu.mt

Matthew Sacco
Research & Development Unit
SeyTravel Ltd.
Birkirkara, Malta
matthew@seytravel.com

***Abstract*—In modern computer vision models, the quality and quantity of training data have become crucial. Datasets deemed sufficient a few years ago now require data augmentation to increase their size. This presents a challenge, especially when these supplementary datasets lack annotations in standard formats like COCO, VGG, or YOLO. One solution to this problem is to learn semantic boundaries from binary images of unannotated datasets, thereby increasing the data available for training and evaluating models. However, choosing an efficient annotation method can be both time-consuming and effort-intensive. This research paper explores three approaches, ranging from traditional image processing algorithms to the recently introduced Segment Anything Model (SAM). The study demonstrates how these different algorithms perform on various datasets and concludes that the proposed image processing method strikes the best balance between performance and efficiency.**

***Index Terms*—Computer Vision, Data Annotation, Datasets**

I. Introduction

In the rapidly advancing area of computer vision, the quality and volume of training data have become increasingly critical for the performance of machine learning models [1]. The need for more diverse and extensive training data has become increasingly apparent, especially when dealing with complex tasks such as object detection and segmentation [2]. This situation is made more challenging when the available datasets lack annotations in standard formats like COCO [3], VGG [4], or YOLO [5].

A promising approach to address this is to utilize binary images from unannotated datasets to expand the data pool for training and evaluating models [6]. For example, the COTS dataset [7], aimed at evaluating inpainting techniques [8], contains colored images with unannotated binary masks.

The process of selecting an efficient annotation method can be time-consuming, mainly if effort is made to optimize it. This paper proposes two algorithms that facilitate the automatic annotation of such datasets and compares them to the current state-of-the-art segmentation model [9].

It is followed by an extensive evaluation of the three techniques on the COTS and COCO datasets to facilitate the choice of an approach to such annotation automation. Our findings suggest that the proposed image processing approach offers the best balance between performance and efficiency, providing a valuable contribution to the field of computer vision and dataset annotation.

II. Background

The COCO (Common Objects in Context) [3] dataset is widely used for object detection, segmentation, and captioning tasks in the field of computer vision. It uses a specific annotation format that includes information such as object category and its location information using bounding boxes or segmentation masks. The COCO annotation format is relevant for several reasons. First, it provides a standardized format for different tasks and models, facilitating comparison and benchmarking. Second, the COCO annotation format includes rich information that can help models learn more complex representations and achieve better performance [10]. Third, the COCO annotation format is supported by many popular computer vision libraries and tools, making it easily accessible and widely adopted [11].

Visual Geometry Group (VGG) [4] is a standard deep convolutional network (CNN) architecture that is widely used as a benchmark and applied to different applications such as facial analysis [12]. Due to its popularity, its annotation format is also widely used in a number of datasets.

The YOLO (You Only Look Once) computer vision annotation format is a text-based format used to annotate objects in an image for training YOLO-based object detection models. Each annotation line contains the object's class label, center coordinates, width and height of the bounding box, all represented as relative values with respect to the image dimensions. This straightforward format defines object locations and classes

979-8-3503-9576-1/23/$31.00 ©2023 IEEE
DOI 10.1109/ISM59092.2023.00049

to train YOLO for real-time object detection tasks. YOLO-based object detectors are widely used due to their ease of customization [5], so their annotation format is also popular.

III. Related Work

Image segmentation is a computer vision task concerning assigning labels to objects in images at a pixel level.

Image processing approach – Active contour models use an image processing approach to fit a deformable curve [13], known as a "snake" energy function, to the contours of an object of interest in the image. This function considers image features to attract the curve towards object boundaries while maintaining smoothness [14].

Clustering approach – K-means clustering is an unsupervised technique used for image segmentation. By grouping pixels with similar color or intensity into clusters, it partitions the image into distinct regions. K-means iteratively assigns pixels to clusters and sets cluster centers to the average pixels within each cluster until convergence. The result is a segmented image into K groups of similar pixels. [15].

The **Vision Transformer (VT) approach** is a novel approach that follows CNNs. It uses self-attention to understand an image's global contexts. VTs process image sequences of tokens and learn important features directly from the pixel data for image segmentation. Segment Anything Model (SAM) [9] is a state-of-the-art VT foundation model that can segment images and generate masks of objects.

IV. Methodology

This paper proposes two approaches to generate polygon annotations from binary masks automatically. These methods have been designed and optimized in the context of existing datasets with binary masks but are not annotated in a standard approach that facilitates augmentation with other major datasets. The code for the proposed techniques is available as open-source on a dedicated repository named `mask-to-annotation`[1].

A. Polygon Approximation Approach

In the first approach outlined in Algorithm 1, Gaussian blurring is applied to the original mask as a form of anti-aliasing to account for masks that may be lower in quality than the corresponding color image, as well as prevent jagged edges. Erosion is then applied as a form of closing to fill in any gaps which could cause overfitting. The Ramer-Douglas-Pecker algorithm [16], $approxPolyDP(c, \epsilon)$ in Algorithm 1, is applied to each contour found in the mask to approximate the polygon surrounding the mask and reduce excessive verbosity and further overfitting of the annotation.

This polygon approximation algorithm takes the distance dimension ϵ as a parameter. An experiment estimated the optimal value for ϵ to be 0.005, determined by recording the average Intersection over Union (IoU) metric error between the generated and ground truth masks of 80 images in the COTS dataset for $\epsilon \in \{0.0, 0.001, \ldots, 0.01\}$ as illustrated in Figure 1.

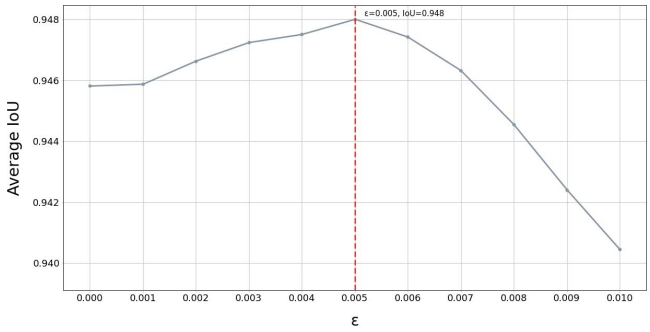

Fig. 1. Results of the hyperparameter optimization experiment – ϵ vs the Average IoU between the original and generated masks across 80 images in the COTS dataset with the best result occurring at 0.005 with an IoU=0.948.

Algorithm 1 Polygon Approximation Approach

Require: $m = \text{mask}$
Require: ϵ
  $m \leftarrow \text{gaussianBlur}(m)$
  $m \leftarrow \text{erode}(m, 3 \times 3)$
  $C \leftarrow \text{findContours}(m)$
  for c in C **do**
    $c \leftarrow \text{approxPolyDP}(c, \epsilon)$
  end for
  return C

B. K-Means Clustering Approach

In the second approach, Gaussian blurring and erosion are also applied. K-means clustering [15] is applied with random initial centers to approximate the cluster centers given the contours found in the mask. The contours are represented as data points in a feature space, and the algorithm attempts to discover the k cluster centers that best reflect the contours. A convex hull is created from the approximated centers, which serves as the polygon annotation of the inputted object mask.

C. Multiple Objects

When images contain more than one object, the above algorithms can still be used. In this case, the number of objects demarked as components will be identified in the first part of the process, and Algorithm 1 and clustering will be executed iteratively over each object.

D. Bounding Box Annotation/YOLO Format

The bounding box annotation method is defined by four parameters, the coordinates of the top-most left corner, and the width, and height. This annotation is computed on binary masks by taking a list of contours as input and calculating the smallest bounding rectangle encompassing all the points. Figure 6 illustrates an example of this annotation style.

[1] `https://github.com/dylanseychell/mask-to-annotation`

V. Evaluation

A. Datasets

Two datasets of a similar structure were chosen to evaluate these approaches. The widely used COCO dataset was used since one of the objectives of this work is to return annotations that match its standard. The smaller and less popular dataset COTS was then used to demonstrate that these approaches can also be used to augment smaller datasets. COTS is an RGB-D dataset featuring images of objects against a green screen, accompanied by binary masks [7]. These masks served as the ground truth to evaluate the automatic annotation presented. The proposed techniques were also tested on a 168-image subset of the COCO 2017 training dataset, containing annotations that could be converted into masks and used as input to the techniques.

B. Metrics

The primary evaluation metric used was Intersection over Union (IoU) in Equation 1. IoU indicates the similarity between the ground truth and the generated annotation when converted back into a mask by considering the masks' spatial alignment and disregarding differences in hue/intensity. Compactness, in Equation 2, was used to gauge how closely the annotations wrap around the original masks when compared to one another, with a lower compactness value denoting a more compact annotation.

$$IoU(GT, A) = \frac{|GT \cap A|}{|GT \cup A|} \tag{1}$$

$$C(A) = \frac{\text{Area}(A)}{\text{Area}(A_{\text{b\_box}})} \tag{2}$$

The average runtime per image for each technique on the COTS dataset was also recorded to demonstrate their usability when applied to large datasets.

C. Results

As applicable, the proposed techniques were evaluated on an Intel Core i5-9400F CPU and an NVIDIA GeForce RTX2060 GPU. The Average IoU and Average Compactness across the datasets were measured for each experiment. The results are presented in Table I. While the Segment Anything algorithm slightly outperformed the other techniques in terms of Average IoU on the COTS Dataset, the proposed Polygon Approximation approach outperforms the other techniques in all other instances. From a performance perspective, the proposed Polygon Approximation method executes in significantly less time on a CPU than the other techniques. These results also show how the Segment Anything Method is optimized to perform better on a GPU, though it is still slower than when the proposed techniques run on a CPU.

D. Visual Results

This section presents a selection of visual results of the three techniques being evaluated. Figures 2 and 3 show a sample from the COTS and COCO datasets, respectively, that are used in the other figures. Figures 4, 5 and 7 present the polygon annotation results of the three techniques on the same samples. This shows that while the proposed Polygon Approximation method successfully wraps around the masks, the other techniques fail to do so in the exact instances. Figure 6 illustrates a visual demonstration of how the same technique can be used to generate a bounding box.

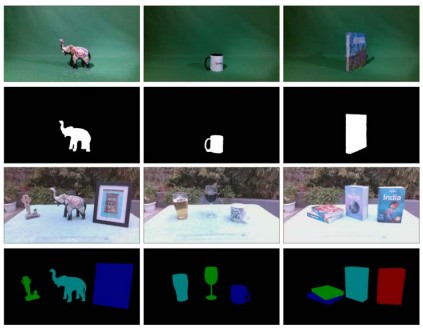

Fig. 2. Sample images featuring from the COTS Dataset and their corresponding binary masks (multiple objects differentiated using color masks).

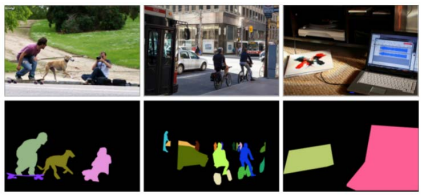

Fig. 3. Sample images featuring from the COCO Dataset and their corresponding binary masks (multiple objects differentiated using color masks).

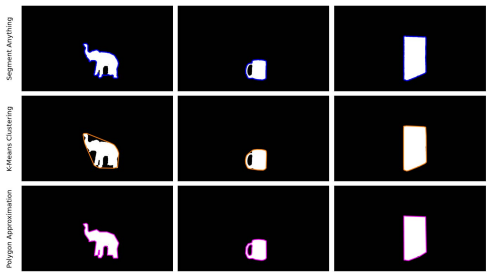

Fig. 4. Results of carrying out polygon annotation using the three different algorithms on the masks featuring single objects in Figure 2. From top to bottom: Segment Anything, K-Means Clustering, Polygon Approximation.

Manual tests were performed on the generated annotation files to fully verify the accuracy of the annotation, as well as conformity to the corresponding annotation formats. The tests also verified the correct assignment of contour labels to their individual object categories in the case of multiple object mask annotation, carried out on makesense.ai.

VI. Conclusion

This paper proposed two approaches for automatically generating polygon annotations using binary masks. The Polygon

Technique	Avg. IoU		Avg. Compactness		Avg. Runtime/image (s)	
	COTS	COCO	COTS	COCO	CPU	GPU
Segment Anything	**0.978**	0.579	0.720	0.599	361.4177	13.684
K-Means Clustering	0.854	0.755	0.796	0.706	0.0382	*N/A*
Polygon Approximation	0.947	**0.815**	**0.716**	**0.553**	**0.0163**	*N/A*
Average	0.926	0.716	0.744	0.619	120.4907	*N/A*

TABLE I
EVALUATION RESULTS OF THE THREE TECHNIQUES ON THE COTS AND COCO DATASETS WITH THE RESPECTIVE METRICS AND THEIR PERFORMANCE. THE BEST-PERFORMING RESULTS ARE DENOTED IN BOLD.

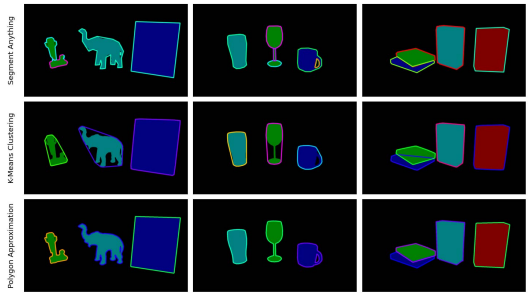

Fig. 5. Results of carrying out polygon annotation using the three different algorithms on the masks featuring multiple objects in Figure 2. From top to bottom: Segment Anything, K-Means Clustering, Polygon Approximation.

Fig. 6. Result of carrying out bounding box annotation on the masks.

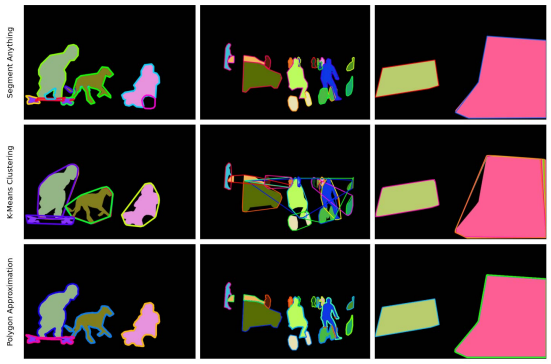

Fig. 7. Results of carrying out polygon annotation using the three different algorithms on the masks in Figure 3. From top to bottom: Segment Anything, K-Means Clustering, Polygon Approximation.

Approximation and K-Means Clustering algorithms are evaluated on the COTS dataset and a subset of the COCO dataset. The proposed Polygon Approximation approach outperforms the K-Means Clustering and Segment Anything algorithms in both metrics, achieving accurate and efficient segmentation. The results also show that the proposed method is optimized to generate high-quality annotations with minimal use of computational resources, thus facilitating the availability of more data for computer vision tasks.

REFERENCES

[1] Keith Man and Javaan Chahl, "A review of synthetic image data and its use in computer vision", *Journal of Imaging*, vol. 8, no. 11, 2022.

[2] S. M. Siamus Salahin, M. D. Shefat Ullaa, Saif Ahmed, Nabeel Mohammed, Taseef Hasan Farook, and James Dudley, "One-stage methods of computer vision object detection to classify carious lesions from smartphone imaging", *Oral*, vol. 3, no. 2, pp. 176–190, 2023.

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár, "Microsoft coco: Common objects in context", 2015.

[4] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition", *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 9 2014.

[5] Muhammad Hussain, "Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection", *Machines*, vol. 11, no. 7, 2023.

[6] Zihan Yang, Richard O. Sinnott, James Bailey, and Qiuhong Ke, "A survey of automated data augmentation algorithms for deep learning-based image classification tasks", *Knowledge and Information Systems*, vol. 65, no. 7, pp. 2805–2861, Jul 2023.

[7] Dylan Seychell, Carl James Debono, Mark Bugeja, Jeremy Borg, and Matthew Sacco, "Cots: A multipurpose rgb-d dataset for saliency and image manipulation applications", *IEEE Access*, vol. 9, pp. 21481–21497, 2021.

[8] Dylan Seychell and Carl J. Debono, "An approach for objective quality assessment of image inpainting results", in *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*, 2020, pp. 226–231.

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick, "Segment anything", *arXiv:2304.02643*, 2023.

[10] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari, "Coco-stuff: Thing and stuff classes in context", in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1209–1218.

[11] Shunjun Wei, Xiangfeng Zeng, Qizhe Qu, Mou Wang, Hao Su, and Jun Shi, "Hrsid: A high-resolution sar images dataset for ship detection and instance segmentation", *IEEE Access*, vol. 8, pp. 120234–120254, 2020.

[12] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman, "Deep face recognition", in *British Machine Vision Conference*, 2015.

[13] Richard Szeliski, "Computer vision algorithms and applications", 2011.

[14] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, "Snakes: Active contour models", *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, January 1988.

[15] Nameirakpam Dhanachandra, Khumanthem Manglem, and Yambem Jina Chanu, "Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm", *Procedia Computer Science*, vol. 54, pp. 764–771, 2015.

[16] David H Douglas and Thomas K Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.