HAISTA-NET: Human Assisted Instance Segmentation Through Attention

Muhammed Korkmaz
Koc University
Istanbul, Turkey
mkorkmaz20@ku.edu.tr

T. Metin Sezgin
Koc University
Istanbul, Turkey
mtsezgin@ku.edu.tr

Abstract

Instance segmentation is a form of image detection which has a range of applications, such as object refinement, medical image analysis, and image/video editing, all of which demand a high degree of accuracy. However, this precision is often beyond the reach of what even state-of-the-art, fully automated instance segmentation algorithms can deliver. The performance gap becomes particularly prohibitive for small and complex objects. Practitioners typically resort to fully manual annotation, which can be a laborious process. In order to overcome this problem, we propose a novel approach to enable more precise predictions and generate higher-quality segmentation masks for high-curvature, complex and small-scale objects. Our human-assisted segmentation model, HAISTA-NET, augments the existing Strong Mask R-CNN network to incorporate human-specified partial boundaries. We also present a dataset of hand-drawn partial object boundaries, which we refer to as "human attention maps." In addition, the Partial Sketch Object Boundaries (PSOB) dataset contains hand-drawn partial object boundaries which represent curvatures of an object's ground truth mask with several pixels. Through extensive evaluation using the PSOB dataset, we show that HAISTA-NET outperforms state-of-the art methods such as Mask R-CNN, Strong Mask R-CNN, and Mask2Former, achieving respective increases of +36.7, +29.6, and +26.5 points in AP_{Mask} metrics for these three models. We hope that our novel approach will set a baseline for future human-aided deep learning models by combining fully automated and interactive instance segmentation architectures.

1. Introduction

In recent years, demand has grown for deep-learning-based, fully automated instance segmentation models [18, 6, 14, 20, 9, 31, 13, 36, 39]. Due to rapid progress in their development, these image detection tools have become the preferred method for applications such as object refinement [8], medical image screening [37, 38], and image/video editing [3, 21]. However, they often perform poorly when objects are small-scale or have a pronounced curvature, causing inaccuracies in the instance segmentation mask, such as boundary outflows or under-covering [35]. To improve mask precision, users may resort to manual annotation [2], though this has several drawbacks including the added cost of time.

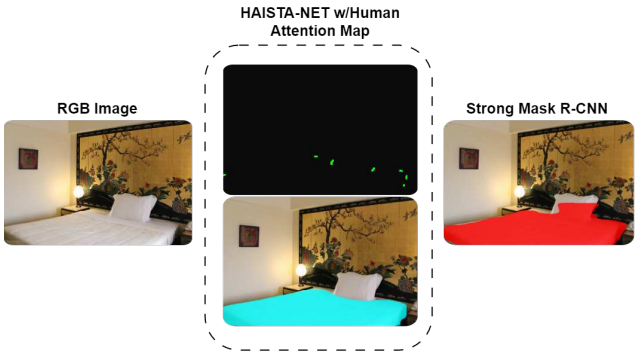


Figure 1: First glance of precise mask prediction by HAISTA-NET using human attention maps dataset (center image). HAISTA-NET outperforms the mask prediction of Strong Mask R-CNN (right) on high-curvature objects.

As a result, researchers have been studying alternative approaches to interactive segmentation [4, 22, 23, 1], to remedy existing deficiencies in segmentation masks while reducing the time needed for manual annotation. One method involves deep-learning-based interactive segmentation models that use corrective action techniques for object mask retrieval [4, 24]. Another proposed technique involves mask editing based on mouse clicks [30], whereby the user helps to guide the automated model manually by detecting boundary outflows or under-covering.

With HAISTA-NET, we bring a new approach to instance segmentation by combining two main areas: fully automated instance segmentation, and interactive instance segmentation. With this proposed method, users convey the intended boundary by marking it manually. A deep learn-

ing network, which has previously been trained with such markings, uses this input to drive the instance segmentation process. Our novel approach eliminates mask errors by training the model with user inputs, to fix the mask where other models stumble and fail.

We also present our new Partial Sketch Object Boundaries (PSOB) dataset. This dataset has images, objects, and categories taken from LVIS [16] and extended with additional hand-drawn sketches. It contains raster images, which we refer to as human attention maps, drawn by users with a few pixels (minimal input) of the high-curvature regions of the object boundary, where segmentation masks are most erroneous. For the PSOB dataset, 30 users annotated 18,677 objects of different scales and a varying number of high curvature sections.

HAISTA-NET architecture uses human attention maps both for the training phase and for running inference. Our model can be easily integrated with various deep learning-based segmentation and detection models (see 3.4). HAISTA-NET uses a Strong Mask R-CNN baseline [18, 15]. In our model we removed the first convolution layer of the Mask R-CNN [18] backbone and added a new one that is fed by a combination of the Human Attention Map and a three-channel RGB image. As a result of our experiments, the HAISTA-NET architecture outperforms the Strong Mask R-CNN baseline with a more precise mask for high-curvature and/or small-scale objects (Figure 1).

We also developed a user-friendly interface that allows users to interact with objects to create or edit human attention maps using partial strokes [29]. By using this interface, users can either directly annotate images without prior knowledge of the segmentation results or perform the annotation after viewing outputs from the Strong Mask R-CNN.

When using the PSOB dataset with human attention maps, our model achieves more accurate results than existing state-of-the-art models. Evaluations conducted with the PSOB dataset show that our model achieves a performance that is 36.7 points better than Mask R-CNN, according to the AP_{Mask} metric, and +29.6 points compared with Strong Mask R-CNN. Using the AP_{Bbox} metric, our model also demonstrates an increase of +33.5 and +31.1 points, respectively, versus Mask R-CNN and Strong Mask R-CNN (see Section 4.1). Moreover, HAISTA-NET achieves a +26.5-point increase in AP_{Mask} versus Mask2Former [9], which is the current state-of-the-art model for instance segmentation on the COCO [27] dataset.

Our main contributions can be summarized as follows:

- We have developed HAISTA-NET based on the Mask R-CNN architecture so that three-channel RGB images can be combined with a human attention map.
- We propose a sketch-based representation of user-defined, high-curvature sections of objects of interest, called the human attention map.
- We present the Partial Sketch Object Boundaries (PSOB) dataset, which will be a valuable asset in driving further research in user-assisted instance segmentation.
- We propose the Adaptive Curvature Number Detector (ACND) for detecting and classifying the curvatures of segmentation masks.
- Applying multiple factor analysis, we have reported our model results for objects according to different scales, different curvature numbers, and different user drawing characteristics.
- Our user study also provides analysis of the cost of sketch-based manual annotations in LVIS dataset quality.

2. Related Work

Instance segmentation methods can be divided into two main categories: fully automated instance segmentation [18, 6, 14, 20, 9, 31, 13, 36, 39], and interactive instance segmentation [4, 22, 23, 1, 3, 21, 24]. The Strong Mask R-CNN [18, 15] tool is one of the state-of-the-art models available that achieves a high degree of precision. This architecture extends from the Faster R-CNN [33] software and sets a baseline for many deep learning models with its simplicity and ease of implementation. With the contributions of Ghiasi *et al*. [15], the simple copy-paste method has improved the performance of the basic Mask R-CNN architecture and transformed it into the Strong Mask R-CNN model. Strong Mask R-CNN, our baseline, became prominent with its practical use, namely by obtaining high precision masks with fast forward training and low memory usage.

Transformer-based architectures [7, 10, 28] are increasing in popularity due to their high image feature learning and attention mechanisms. For example, Mask2Former [9] architecture uses transformers in place of the conventional RPN backbone of Faster R-CNN, and surpasses both standard Mask R-CNN [18] and Strong Mask R-CNN [18, 15] in results with its transformer-based structural design. These new-generation transformers are replacements of CNN-based traditional architectures. Yet such models still have a high memory and time cost [17], not only for training but also for inference. Also, they tend to underperform in fine-grained applications such as medical image analysis [32].

To avoid segmentation failures in object boundaries, some researchers use time-costly manual annotation techniques. Interactive segmentation is another alternative that has emerged recently and promises superior results to manual annotation. Some studies suggest using mouse clicks



Figure 2: **Method Outline.** Users draw a couple of pixels according to their attention to the object. Then, a Human Attention Map is generated to concatenate with the RGB image to feed the model. We denote the concatenate operator as $\otimes$.

or free-style painting [25] to correct the boundaries of segmentation masks, assuming this reduces annotation time.

According to Benenson *et al*. [4], the point and click-based [41] method is an effective technique to improve mask precision rates. Using mouse clicks requires two types of marking for redefining object boundaries. If the estimated segmentation mask overflows the actual boundary of the object, the user sets negative markers to ignore those segments. If the boundary under-covers the ground truth, the user sets the targeted distance by placing positive markers. However, the mouse click technique may fail for small-scale objects as it requires a high level of detail to adjust in bounding box limits.

In this study, instead of using time-consuming mouse clicks to edit faulty masks, we advocate partial marking of the object boundary prior to instance segmentation, and then using these marks to aid segmentation. The boundary is marked specifically at problematic high-curvature regions where the segmentation fails the most.

3. Proposed Approach

We propose a methodology to generate more precise segmentation masks by integrating human attention maps with fully automated segmentation models [18, 15]. Our study includes the following steps: a review of dataset collection, data annotation, selection of input representation techniques, conversion of input representations into human attention maps, demonstration of network architecture, fine tuning of training parameters, and running inference.

3.1. Partial Sketch Object Boundaries Dataset

When first creating our dataset and deciding on which annotation technique to use we explored the failure conditions of different instance segmentation models. First, we trained a Strong Mask R-CNN [18, 15] model with the LVIS dataset to investigate segmentation mask failures. Following this training, we selected 3,070 test images from the LVIS [16] dataset to run inference in order to determine object mask failures. The object mask prediction was not precise in the following conditions:

- If the object scale is small, with an $area < 32^2$.
- If the object is medium-scale $32^2 < area < 96^2$ or large-scale $area > 96^2$ and has a high degree of curvature or number of curves.
- If the object is located behind another object, causing an occlusion problem.
- If the object's shape and curvature differ in test images from training images.

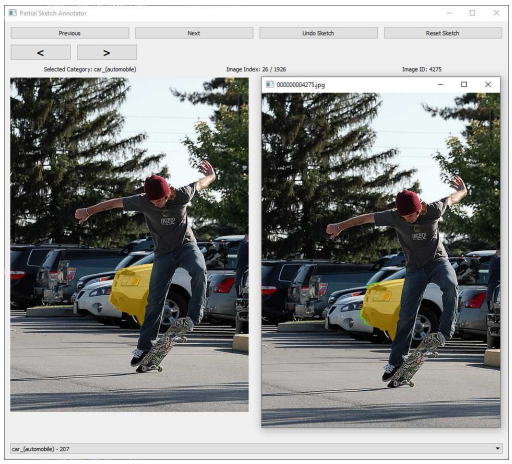


Figure 3: **Interactive interface.** Users can interact with the target object via this tool.

Regarding the analysis results, we propose a minimal sketch-based technique to extend the LVIS dataset with user input. Our newly constituted dataset, PSOB, contains hand-drawn partial object boundaries in high-curvature points (see Section 3.2) for 115 categories of objects taken from the LVIS dataset. Thirty users annotated data, using hand-drawn partial sketches to represent object high-curvature points. Our aim was to minimize human effort while interacting with the object. For the data distribution to be reasonable, we annotated 10,450 training objects, 4,109 validation objects, and 4,118 test objects. For experimental reasons, we also recorded the object's total sketch time [sec],

time for each stroke [sec], and the pixel-wise stroke's corresponding values in the x/y coordinate frame.

The important factor of this dataset is human attention points, which represent user inputs to form human attention maps. To obtain these points, some parts of the boundaries of the objects of interest are sketched by users. These sketched points represent human assistance in our work. We identify this assistance according to the following rules:

- If sketch input covers less than 25% of the object's boundary, human assistance is designated as **minor assistance**.
- If sketch input covers between 25% and 50% of the object's boundary, human assistance is designated as **medium assistance**.
- If sketch input covers more than 50% of the object boundary, human assistance is designated as **major assistance**.

3.2. Adaptive Object Curvature Detector

Object masks consist of polygons having different angles between line segments. Angles of intersecting line segments constitute a high-curvature or low-curvature point depending on their degree. We use the Ramer-Douglas-Peucker (RDP) [11] algorithm to refine curvature points for each object boundaries.

Object masks consist of polygons with different angles between their line segments. Angles of intersecting line segments constitute a high-curvature or low-curvature point depending on their degree. We use the Ramer-Douglas Peucker (RDP) [11] algorithm to refine curvature points for each object boundary.

The standard RDP algorithm uses a static epsilon value to simplify the number of the object's corners. Designated static RDP epsilon values require parameter tuning for every scale of the object. In order to fine tune this value, we calculate the perimeter of the polygon from the coordinates and set the epsilon value to 3% of the perimeter. This method returns more accurate results while calculating the number of curvature points for objects from different scales, compared with classical RDP. We use the following formula, where ϵ is adaptive epsilon for RDP, n is the line segment count, and $\mathcal{I}$ represents line segments:

$$\epsilon = 0.03 \cdot \sum_{k=1}^{n} \sqrt{(\mathcal{I}_{k_{x_2}} - \mathcal{I}_{k_{x_1}})^2 + (\mathcal{I}_{k_{y_2}} - \mathcal{I}_{k_{y_1}})^2} \quad (1)$$

After obtaining the results of the adaptive object curvature detector (eq. 1), we classify the objects according to their number of curvature points by the using following rules:

- If an object's number of curvature points is less than six, an object's curvature type is designated as **low curvature**.
- If an object's number of curvature points is between six and 10, its curvature type is designated as **medium curvature**.
- If an object's number of curvature points is greater than 10, it is designated as a **high curvature** type.

3.3. Representation of Human Attention Map

As briefly described in previous sections, high-curvature points on the objects of interest are partially annotated by human users in order to create human attention maps. The x and y pixel coordinates of these user strokes are used to form binary images with the same dimensions as input images. These binary images are representations of the human attention map and are later supplied to the fourth channel of the first convolution layer, where the other three channels are reserved for the original RGB image. Attention points that are annotated by human users as high curvature points are set to the pixel value of 255 while other regions are left initially as zero. This results in a sparse representation where most of the image is zero while a few pixels at the location of interest are 255. During the training phase, random initialization of the fourth channel weights were used.

During the preliminary analysis stage, the effects of representing unrelated regions with various forms were investigated. Since, during back-propagation, randomly multiplying the initialized weights of the fourth channel with zeros prevents optimization of these weights, alternative strategies were tested, such as using a small fixed value or using a randomly generated small number instead of zeros.

The results show an insignificant difference in performance among the three methods proposed for representing unrelated regions, namely: using zeros, using a small fixed pixel value ($\mathcal{P}$) such as 10, and using a randomly generated number. Because of this, using a fixed pixel value of 10 for regions outside of attention points was selected for the rest of the study (eq. 2).

$$\mathcal{P} = \begin{cases} 255, & \text{if location is attention point} \\ 10, & \text{otherwise} \end{cases} \quad (2)$$

3.4. Network Architecture

HAISTA-NET adds a head branch to the Mask R-CNN [18] architecture. We remove the first convolution from the backbone of Mask R-CNN to set our 2D convolution. We concatenate the 3-channel RGB image with our human attention map to form 4-channel image inputs (Figure 2). We set the new convolution that overrides the first convolution of the current backbone with an input channel parameter equal to 4 and an output channel parameter equal to 64, as represented in standard Res-Net [19] architecture. The new convolution kernel size is 7x7, the stride is 2x2, and the



Figure 4: **Visualization of the Predictions of HAISTA-NET.** We present images with different scales, curvature numbers, and hand-drawing-based assistance types.

padding is 3x3. While adopting the new weights of the 4-channel convolution, for the first three channels, instead of randomly initializing weights, we use pre-trained weights from the existing 3-channel convolution. Because the human attention map channel is new, we randomly initialize the weights between a range of (0, 0.001). As we described earlier, Mask R-CNN [18] architecture extends from Faster R-CNN [33]. The region proposal network backbone of Faster R-CNN uses data transformation parameters of mean and standard deviation vectors to adopt new input values when the channel number increases or decreases. Therefore, we need to adjust the final channel's mean and standard deviation vectors. We set the mean of the final channel as 0.5. The mean vector of first three channels is (0.485, 0.456, 0.406).

Additionally, we set the standard deviation of the final channel to 0.2, where the standard deviation vector for the first three channels is (0.229, 0.224, 0.225). We evaluate Res-Net [19] and Res-NeXt [40] architectures of depth 50, 101, and 152 layers with feature pyramid networks (FPN) [26]. We achieved the best results with Res-NeXt-101-FPN.

3.5. Data Augmentation

In order to develop our enhanced model, we performed different training techniques [12]. Firstly, we flipped the image randomly and assigned the probability value of being flipped to 0.5. As a result of this, the sample diversity increased. Secondly, we used the large-scale jitter augmentation technique to randomly resize the image and image bounding box. We resized the image and boundary within the scale range (0.1, 2). The transform of the target size was 1024x1024. We also used bilinear interpolation as a parameter. Finally, we used the fixed-size crop to scale the image to the target size. Before feeding the model with our dataset, we used the simple copy-paste [15] data augmentation method to create images from rare categories. We took object ground truth annotations from two images, then created a new image using these annotations to reproduce samples. Finally, we compared the original 3-channel Mask R-CNN and our own model regarding the effectiveness of this simple copy-paste data augmentation method.

The results show that Mask R-CNN achieves $\approx$7 AP improvements while we achieve less than $\approx$3 AP.

3.6. Training Parameters

We fine-tuned HAISTA-NET with 26 epochs for both ResNet [19] and ResNeXt [40] backbones using SGD optimizer. The learning rate of the optimizer was 0.0025. We set multi-step learning rate scheduling to decrease the learning rate by 10 at the 16th and 22nd epochs. In addition, we set a weight decay of 0.0001 and momentum of 0.9.

We trained these three different models on the PSOB dataset in order to compare their performances. Since the PSOB dataset contains Human Attention Maps, RGB images, and annotation features such as category, bounding box, and mask polygon, we can apply the same process to these 3 models. However, only HAISTA-NET uses Human Attention Maps due to model configuration. Mask R-CNN and Mask2Former only use RGB images to train.

Since we are presenting a new dataset, PSOB, we must analyze the performance of other popular models such as Strong Mask R-CNN and Mask2Former on this dataset. For Mask R-CNN training, we used the same hyperparameters as HAISTA-NET because our model is based on MASK R-CNN. However, since Mask2Former is a transformer-based architecture, we tuned this model with 25 epochs, and the AdamW optimizer was used with a learning rate of 0.0025. Also, we set a weight decay of 0.05, epsilon of 1e-08, and betas of (0.9, 0.999)

We trained these three different models on the PSOB dataset in order to compare their performances. Since the PSOB dataset contains human attention maps, RGB images,

Model	Backbone	Mask					Bbox				
		AP	AP_{50}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_S	AP_M	AP_L
Mask R-CNN	RES-50 + FPN	18.2	30.3	4.6	15.0	25.7	23.3	34.4	14.0	23.5	27.4
Mask R-CNN w/SCP	RES-50 + FPN	25.3	36.3	12.4	23.0	34.4	25.7	36.6	19.1	27.6	30.6
Mask2Former	RES-50 + FPN	28.4	36.4	13.0	25.7	38.6	27.7	35.0	16.7	27.2	35.1
Mask2Former w/SCP	RES-50 + FPN	30.3	39.4	12.6	27.0	39.5	30.1	38.2	14.0	30.1	35.5
HAISTA-NET	RES-50 + FPN	51.2	68.5	41.8	53.5	57.2	52.1	69.3	51.4	56.8	53.1
HAISTA-NET w/SCP	RES-50 + FPN	51.2	68.9	41.4	53.5	57.7	53.3	69.2	53.4	58.4	54.1
HAISTA-NET	RES-101 + FPN	51.0	69.1	40.0	52.7	60.5	50.5	69.9	45.2	55.5	53.9
HAISTA-NET w/SCP	RES-101 + FPN	52.2	69.8	41.6	54.4	59.2	54.5	70.0	54.5	59.9	55.6
HAISTA-NET	RES-152 + FPN	52.8	71.0	39.8	55.4	60.1	52.6	71.7	49.6	57.3	54.6
HAISTA-NET w/SCP	RES-152 + FPN	53.1	71.3	45.2	56.6	59.4	55.3	71.7	55.4	61.5	54.4
HAISTA-NET	X-101 + FPN	52.0	70.9	40.1	54.9	58.1	51.5	71.4	45.3	57.5	52.7
HAISTA-NET w/SCP	X-101 + FPN	**54.9**	**73.8**	**47.8**	**58.4**	**61.5**	**56.8**	**74.4**	**56.0**	**62.3**	**57.2**

Table 1: The results of the models with different backbones and augmentation techniques. SCP represents the Simple Copy Paste data augmentation [15] technique. We train HAISTA-NET 26 Epochs without Simple Copy Paste and 50 epochs with Simple Copy Paste. The training combinations of Mask R-CNN are same as HAISTA-NET. However, both versions of Mask2Former are trained with 25 epochs.

as well as annotation features such as category, bounding box, and mask polygon, we can apply the same process to these three models. However, only HAISTA-NET uses human attention maps as a result of its model configuration. Mask R-CNN and Mask2Former only use RGB images to train.

PyTorch Framework is used for all implementations and configurations. In all experiments for training, we choose NVIDIA RTX 3090 TI single GPU with a batch size of 2. The image shuffling technique randomly selects images from the dataset in the training phase. We use two subprocesses for data loading in training.

3.7. Inference

We performed inference for 4,118 annotations after the training. This was conducted using NVIDIA RTX 3090 TI with a single GPU with a batch size of 1. We obtained AP [27], AP_{50}, AP_S, AP_M, and AP_L metrics for all models. HAISTANET, which has strong data augmentation [15] and a ResNeXt-101-FPN backbone, results in the best object mask and bounding box predictions.

In order to interact with images during inference time, we present a user-friendly graphical user interface (Figure 3). With the help of the interface, users may upload an image in seconds and create hand-drawn sketches to use as a human attention map. Following the sketch, a user can observe object mask results as well as the bounding box of the object. Model guidance is possible by using the interface in one of two ways. In the first case, the image goes into the Mask R-CNN model for the user to see the results and to create a human attention map. In this case, a user may detect the missing object parts. Users may use this as a reference to create more precise partial sketches for HAISTA-NET. In the second case, the image is not fed into the Mask R-CNN model, and the user creates heuristic sketches without any prior knowledge of the segmentation results. This leaves the HAISTA-NET to make any predictions concerning the human attention map.

4. Experiment

In order to describe the performance of the models, we report our experimental results, which include AP metrics, mask/bounding box predictions, and analysis of the PSOB dataset. Additionally, we present relational analyses of the user's sketch characteristics along with the curvature types and object scales used in the creation of the PSOB dataset.

4.1. Main Results

We evaluated our HAISTA-NET model with 4,118 test annotations of the PSOB dataset. Our results point out significant improvement over state-of-art models such as Strong Mask R-CNN and Mask2Former. Our analysis demonstrates apparent progress on mask precision deficiencies, such as object boundary outflow or under-covering. Results exhibit the effectiveness of a human attention map to overcome the mask prediction issues that standard models face with small-scale and high-curvature objects. Our experiments support the potential of our architecture for acquiring high precision segmentation masks (Figure 4). We demonstrate that HAISTA-NET produces better results than other models in all AP, AP_{50}, A_S, AP_M, and AP_L results. Comparing the Res-Net-50FPN, Res-Net-101-FPN,

Res-Net-152-FPN, and Res-NeXt101-FPN backbones, we achieved the best performance with Res-NeXt-101-FPN. (see Table 1). We also report that AP values improve if our model uses the simple copy-paste data augmentation method.

We benchmark the HAISTA-NET Strong Mask R-CNN baseline by utilizing output images. HAISTA-NET demonstrates better results with challenging high-curvature objects and small-scale objects (see Table 2).

4.2. Multiple Factor Analysis

Using multiple factor analysis [34], we classified the object given in test data according to scale, curvature, and level of assistance. In order to perform the analysis, we first ran the inference with HAISTA-NET. Later on, we retrieved the mask predictions and split results into classified partitions such as small-scale objects with low-curvature and major user assistance, and large-scale objects with medium curvature and minor user assistance.

We conducted a factorial analysis of variance (ANOVA, two-way) to compare the main effects on size, curvature, and user assistance. We also compared the correlated interaction effects on size $\otimes$ curvature, size $\otimes$ assistance, and curvature $\otimes$ assistance. A statistically significant difference exists between size, curvature, and assistance at p-value $(p) < 0.05$. The main effect of size (F (2,5) = 123.59, p = 0.001), curvature (F (2,5) = 9.13, p = 0.021), and assistance (F (2,5) = 9.21, p = 0.021) in AP_{Mask} are significant such that objects having large size, low curvature, and major or medium assistance receive higher scores. In AP_{Mask} analysis, there is a significant interaction between size and curvature (F (4,5) = 14.05, p = 0.006) as well as curvature and assistance (F (4,5) = 5.28, p = 0.048). The main effect of size (F (2,5) = 16.04, p = 0.007), curvature (F (2,5) = 6.23, p = 0.044), and assistance (F (2,5) = 9.87, p = 0.018) in $AP_{Bbox}$are significant such that objects with a large or medium size, high or low curvature, and minor or medium assistance received higher scores. For AP_{Bbox} analysis, there is a significant interaction between size and curvature (F (4,5) = 15.82, p = 0.005)), size and assistance (F (4,5) = 10.54, p = 0.012), as well as curvature and assistance (F (4,5) = 7.16, p = 0.027) (Figure 5).

4.3. Curvature-Based Average Precision

We report the results of the average precision (AP) values of objects in all scales according to curvature classification. In order to analyze the results of curvature-based AP, we divide the objects of the PSOB test set into low curvature, medium curvature, and high curvature types. In addition, we compare HAISTA-NET AP results with Strong Mask R-CNN for $AP_{low\text{-}curvature}$, $AP_{medium\text{-}curvature}$ and $AP_{high\text{-}curvature}$, demonstrating that HAISTA-NET is more successful than Strong Mask R-CNN for all curvature types (see Table 2).

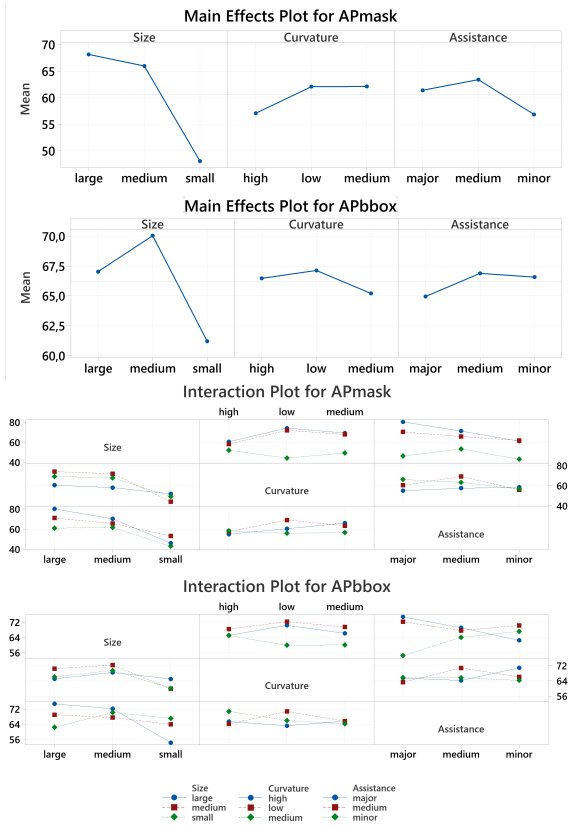


Figure 5: The graphs demonstrate the main and interaction effects of AP_{Mask} and AP_{Bbox} values generated using HAISTA-NET.

Model	Mask		
	$AP_{low\text{-}curvature}$	$AP_{med.\text{-}curvature}$	$AP_{high\text{-}curvature}$
M	25.5	21.4	21.4
H_1	58.1	53.3	50.4
H_2	**61.2**	**58.2**	**51.9**

Table 2: Results of the AP values of the models in different curvature types. M is Mask R-CNN w/SCP + Res-50-FPN, H_1 is HAISTA-NET w/SCP + Res-50-FPN, and H_2 is HAISTA-NET w/SCP + X-101-FPN

4.4. PSOB Interaction Time Analysis

We annotated 18,677 objects according to 115 categories and three different scales (small, medium, and large). This data was collected from 30 users. We stored critical numerical metrics such as total interaction time, area, perimeter (length) of segmentation polygon (mask), length of sketch input, percentage of covering object boundaries with the sketch, stroke count, and sketching time except latency. One

of the most crucial factors of this analysis is the time benchmark. Since our model takes extra user input during training and inference, shorter interaction is better. We developed sketch-based interaction considering this situation. Time analysis shows that total interaction time is roughly three times longer than sketching time because interaction time has latency. The human reasoning process causes an extension of time for detecting an object's exact boundaries. In addition, we performed multiple regression [5] analysis to determine which factors affect total interaction time. We set the total interaction time as a response (dependent variable) and the object's curvature number, length of segmentation polygon, stroke count, and length of sketch input as predictors (independent variables). As a result, all the independent variables significantly predict total interaction time (Table 3) because the p values of all independent variables are less than 0.05. Furthermore, this regression model is statistically significant with R^2 of 76.27%.

	Train	Validation	Test
Interaction Time [sec]	7.2	8.3	8.4
Sketching Time [sec]	2.0	3.2	3.3
No. Of Curvatures	7.6	7.5	8.0
Perimeter of Polygon	695.4	679.6	688.2
LS/PP (%)	19.9	31.2	31.2
Stroke Count	6.5	6.2	6.0
Annotation Count	10450	4109	4118

Table 3: PSOB dataset stores the pieces of information about sketched objects. LS/PP represents "Length of Sketch" over "Perimeter of Polygon." This ratio represents the number of pixels that are drawn on the object's boundaries as a percentage.

4.5. User Study

Users can annotate objects manually if the model output is not satisfying. However, manual annotation is time-consuming compared with PSOB's partial annotation. In particular, retrieving high-detail object polygons such as LVIS [16] annotations requires extra attention and focus. Since LVIS contains fine-grained polygons of object boundaries, we were curious about the interaction time needed for annotating an object's mask with hand-drawn sketches. In order to perform this experiment, we collected 643 objects of different sizes (small, medium, and large) to analyze the average interaction time and mIOU values of LVIS-Like sketches and rough sketches. We then compared the results with the PSOB average interaction time (Table 4) and mIOU value (Table 5). The following steps are specified for performing the study:

- Users draw the boundaries of ground truth masks with fine-grained characteristics such as LVIS annotations, and interaction time [sec] is recorded.
- Users draw the boundaries of ground truth masks as a rough sketch and interaction time [sec] is recorded.
- We feed the model with a human attention map from the PSOB dataset to determine how close our model results are to the LVIS mask.

Results can be summarized as follows;

- LVIS-Like interactions have the best quality but take the longest time and require the greatest effort.
- Rough sketches are fast but do not cover the curvatures of an object due to its characteristics.
- Since PSOB annotations are partial data, they take the shortest time. As a result, we acquire mask qualities closest to LVIS-Like, without complete manual annotation, when we feed the model with the human attention map.

Object		Average Sketch Time [sec]		
Scale	Number	LVIS-Like	Rough	PSOB
small	27	257	15	9
medium	199	116	17	8
large	238	117	16	9

Table 4: Results of the average sketch time differences of LVIS-Like, Rough, and PSOB sketches for different scale objects.

	mIOU		
Interaction Type	Small	Medium	Large
LVIS-Like	99%	97%	96%
Rough	91%	82%	78%
PSOB	98%	93%	91%

Table 5: Results of the mean intersection over union (mIOU) values of LVIS-Like, Rough, and PSOB sketches for different scale objects.

5. Conclusion

Finding the correct boundaries for small-scale objects and objects with high curvature points is a challenging task in instance segmentation. In this study, we present a novel

approach, HAISTA-NET, and the concept of human attention maps, which are shown to achieve significant improvements in these areas compared with current fully automated state-of-the-art algorithms. Our method requires a minimal amount of input from users and does not require longer training and inference time in any noticeable manner. Moreover, we also present a new dataset, PSOB (Partial Sketch Object Boundaries), that combines human attention maps with the LVIS dataset for instance segmentation. Our user-friendly interface brings a new perspective to annotation and interaction techniques. We also provide an extensive analysis of the factors that affect the performance of our architecture and state-of-the-art methods. Multiple factor analysis, according to AP_{Bbox} and AP_{Mask} metrics, shows the most affecting factors.

HAISTA-NET is easy to use and can be extended to other computer vision tasks such as object detection, panoptic segmentation, etc. It can be easily implemented using other visual software applications that use CNNs with a minimal amount of user input. We hope that the evidence we provide in this study encourages other authors to incorporate HAISTA-NET in their own research to improve performance.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, 2018. 1, 2

[2] Mykhaylo Andriluka, Jasper RR Uijlings, and Vittorio Ferrari. Fluid annotation: a human-machine collaboration interface for full image annotation. In *ACM Multimedia*, 2018. 1

[3] Arnaud Benard and Michael Gygli. Interactive video object segmentation in the wild. *arXiv preprint arXiv:1801.00269*, 2017. 1, 2

[4] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *CVPR*, 2019. 1, 2, 3

[5] William D Berry, William D Berry, Stanley Feldman, and Dr Stanley Feldman. *Multiple regression in practice*. Sage, 1985. 8

[6] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019. 1, 2

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[8] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *CVPR*, 2018. 1

[9] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[11] 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*, 1973. 4

[12] Xianzhi Du, Barret Zoph, Wei-Chih Hung, and Tsung-Yi Lin. Simple training strategies and model scaling for object detection. *arXiv preprint arXiv:2107.00057*, 2021. 5

[13] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *ICCV*, 2019. 1, 2

[14] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *ICCV*, 2021. 1, 2

[15] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021. 2, 3, 5, 6

[16] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 2, 3, 8

[17] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. In *NeurIPS*, 2021. 2

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2, 3, 4, 5

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 5

[20] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, 2020. 1, 2

[21] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Transactions on Graphics (ToG)*, 2004. 1, 2

[22] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *CVPR*, 2018. 1, 2

[23] JunHao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In *ICCV*, 2017. 1, 2

[24] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, Sim-Heng Ong, and Jiashi Feng. Multiseg: Semantically meaningful, scale-diverse segmentations from minimal user input. In *ICCV*, 2019. 1, 2

[25] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 3

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 5

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 6

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2

[29] Luke Olsen, Faramarz F Samavati, Mario Costa Sousa, and Joaquim A Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 2009. 2

[30] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *ICCV*, 2017. 1

[31] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *CVPR*, 2021. 1, 2

[32] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps: Automation of Decision Making*, 2018. 2

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2, 5

[34] Lars St, Svante Wold, et al. Analysis of variance (anova). *Chemometrics and Intelligent Laboratory Systems*, 1989. 7

[35] Chufeng Tang, Hang Chen, Xiao Li, Jianmin Li, Zhaoxiang Zhang, and Xiaolin Hu. Look closer to segment better: Boundary patch refinement for instance segmentation. In *CVPR*, 2021. 1

[36] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. Boxinst: High-performance instance segmentation with box annotations. In *CVPR*, 2021. 1, 2

[37] Guotai Wang, Wenqi Li, Maria A Zuluaga, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Interactive medical image segmentation using deep learning with image-specific fine tuning. *IEEE Transactions on Medical Imaging*, 2018. 1

[38] Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Deepigeos: a deep interactive geodesic framework for medical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 1

[39] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *NeurIPS*, 2020. 1, 2

[40] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 5

[41] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *CVPR*, 2016. 3