

METHOD AND SYSTEM FOR TRAINING INSTANCE SEGMENTATION MODEL

CROSS-REFERENCE TO RELATED APPLICATIONS

5 [0001] This patent application claims priority to Korean Patent Application No. 10-2025-0016809, filed on 10 February 2025, with the Korean Intellectual Property Office, the entire contents of which are hereby incorporated by reference.

TECHNICAL FIELD

10 [0002] The disclosure relates to instance segmentation model training, and more particularly, to a training method and system for implementing an instance segmentation model capable of ensuring equal loss contribution for different sized components
15 that constitute a single instance.

BACKGROUND

[0003] Instance segmentation involves detecting distinct instances within an image and segmenting the pixels associated
20 with each instance.

[0004] In images with complex environments, instance segmentation emphasizes handling occlusion between instances.

[0005] Complex environments may feature closely located or overlapping instances.

25 [0006] Instance segmentation in such environments with

overlapping instances is known as occluded instance segmentation.

[0007] Instance segmentation finds broad applications in various fields, including security, industry, and retail, where occlusion scenarios are common, making the resolution of occluded instance segmentation crucial.

[0008] Occluded instance segmentation is a highly challenging computer vision problem, with extensive research focused on developing effective models.

[0009] An ideal instance segmentation algorithm should accurately segment each instance pixel (accuracy), recognize all components related to a single instance as the same instance (completeness), and disregard ambiguous situations such as reflections or crowd blobs (uncertainty).

[0010] Current instance segmentation models often prioritize accuracy while neglecting completeness and uncertainty handling. However, completeness is particularly vital for safety applications.

[0011] For example, in collaborative robots (cobots) that operate alongside humans in shared workplaces, incomplete instance segmentation of human body parts can lead to severe injuries or fatalities.

[0012] Existing models use losses applied uniformly across the entire instance, such as binary cross-entropy (BCE) loss. This approach favors predicting large components but inherently overlooks semantically important small components.

[0013] Therefore, there is a need for an instance segmentation method that effectively applies the training loss to both large and small components.

5

SUMMARY

[0014] The present disclosure is directed to a method for applying training loss effectively to both large and small components during instance segmentation.

[0015] The present disclosure is also directed to a component-
10 balance focal (CBF) loss calculation method that applies loss by focusing on the components of an instance.

[0016] The present disclosure is also directed to a training method and a system for implementing an instance segmentation model that is capable of ensuring equal loss contribution for
15 differently sized components constituting an instance.

[0017] The present disclosure is also directed to a training method and system for implementing an instance segmentation model that is capable of improving completeness in instance segmentation.

20 **[0018]** The present disclosure is also directed to a training method and a system for implementing an instance segmentation model that is capable of ensuring equal loss contribution for differently sized components constituting an instance.

[0019] According to one aspect of the subject matter described
25 in this application, an instance segmentation model training

method can include applying image data to the instance segmentation model to thereby detect an instance mask, calculating a loss based on the detected instance mask and a ground truth (GT) instance mask corresponding to the image data,
5 and updating the instance segmentation model based on the calculated loss, where the loss can be calculated by applying a component-wise weight to a binary entropy loss calculation formula.

[0020] In some implementations, the weight can be set based on a
10 proportion of a component in the mask. In some implementations, the weight may not be applied to an area corresponding to a background and can be applied to an area corresponding to a component of an instance. In some implementations, the loss can be calculated based on an entire mask or sampled points.

[0021] According to another aspect of the subject matter
15 described in this application, an instance segmentation model training system can include a first database configured to store an image dataset, a second database configured to store a ground truth (GT) instance mask set corresponding to the image dataset,
20 and a processor. The processor can be configured to apply image data to the instance segmentation model to thereby detect an instance mask, calculate a loss based on the detected instance mask and a GT instance mask corresponding to the image data, and update the instance segmentation model based on the calculated
25 loss, where the loss is calculated by applying a component-wise

weight to a binary entropy loss calculation formula.

[0022] In some implementations, the processor may not apply the weight to an area corresponding to a background and can apply the weight to the points of an area corresponding to a component of an instance. In some implementations, the processor can calculate the loss based on an entire mask or sampled points.

[0023] According to implementations of the features described in this application, a method in instance segmentation to effectively apply the training loss to both large and small components can provide technical benefits.

[0024] For example, the features can be advantageous in providing a component-balance focal (CBF) loss calculation method to apply loss with a focus on the components of an instance.

[0025] The features can also be advantageous in providing a training method and system for implementing an instance segmentation model ensuring equal loss contribution for different sized components constituting an instance.

[0026] The features can also be advantageous in terms of improving completeness in instance segmentation and instance segmentation performance for both large and small components by training an instance segmentation model using CBF loss.

BRIEF DESCRIPTION OF THE DRAWINGS

[0027] FIG. 1 is a diagram illustrating an example of an instance segmentation model training system.

[0028] FIG. 2 is a flowchart illustrating an example of an instance segmentation model training method.

[0029] FIG. 3 is a graph representing an example of the component-level recall relative to the ratio of component area
5 to instance area.

[0030] FIG. 4 is a diagram illustrating an example of a comparison between the training results of the instance segmentation model and the existing instance segmentation model with the ground truth (GT) instance mask.

10

DETAILED DESCRIPTION

[0031] FIG. 1 is a diagram illustrating an example of an instance segmentation model training system 100.

[0032] With reference to FIG. 1, the instance segmentation model
15 training system 100 (hereinafter referred to as the "system") can provide a component-balance focal loss calculation method that applies loss with a focus on the components of an instance.

[0033] In some implementations, the system 100 can train an instance segmentation model to ensure equal loss contribution for
20 different-sized components constituting a single instance.

[0034] In some implementations, the system 100 can include a first database 110, a second database 120, and a processor 130, and the configuration of the system 100 is not limited thereto.

[0035] In some implementations, the first database 110 can store
25 a plurality of image data used for training the instance

segmentation model.

[0036] Hereinafter, the plurality of image data used for training is referred to as the image dataset.

[0037] For example, the image dataset can include large-scale
5 open-source datasets, and the data including the image dataset is not limited to open-source datasets.

[0038] In some implementations, the second database 120 can store ground truth (GT) data corresponding to the image dataset stored in the first database 110.

10 **[0039]** For example, the ground truth data can refer to the desired answers that the instance segmentation model aims to predict, and the performance of the instance segmentation model can be considered excellent when its output matches the ground truth data with high accuracy.

15 **[0040]** For example, the ground truth data can be labeled and stored in the second database 120, and the plurality of ground truth data stored in the second database 120 can refer to the ground truth dataset.

[0041] For example, the ground truth dataset can include
20 instances that serve as a reference for comparing the segmentation results for instances included in the image data stored in the first database 110.

[0042] Hereinafter, the output of the instance segmentation model may refer to the detected instance mask, and instances stored in
25 the second database 120 for comparison with the output of the

instance segmentation module (detected instance mask) may refer to the GT instance masks.

[0043] One image can include one or multiple instances.

[0044] In some implementations, the processor 130 can include an
5 instance segmentation model and can train the instance segmentation model based on an image data set.

[0045] The processor 130 can compare the detected instance mask, the output of the instance segmentation model, with the corresponding GT instance mask to calculate the loss.

10 **[0046]** The processor 130 can apply the calculated loss to train the instance segmentation model to improve the performance of the instance segmentation model.

[0047] The processor 130 can include a training module 131 and a loss calculation module 132.

15 **[0048]** The training module 131 can include the instance segmentation model and train the instance segmentation model based on the input image dataset.

[0049] The instance segmentation model can detect instances in images through training and output the detected instance mask.

20 **[0050]** The loss calculation module 132 can calculate the loss based on the detected instance mask input from the training module 131 and the GT instance mask stored in the second database 120.

[0051] The loss calculation module 132 can calculate the loss by comparing the detected instance mask with the corresponding GT
25 instance mask.

[0052] The loss calculation module 132 can calculate the newly proposed component-balance focal (CBF) loss by applying component-wise weights to the existing binary cross-entropy (BCE) loss.

5 **[0053]** The loss calculation module 132 can provide the calculated loss to the training module 131, enabling the training module 131 to train the instance segmentation model by reflecting the loss.

[0054] In some implementations, the loss (L_{CBF}) can be expressed as Equation 1.

10 **[0055]** [Equation 1]

$$\mathbf{[0056]} \quad L_{CBF} = \sum_p^A w_p' \cdot l_{bce,p}$$

[0057] Here, w_p' can refer to the weight coefficient, w_{av} can refer to the mean coefficient uniformly applied to the entire mask area, $l_{bce,p}$ can refer to the BCE loss for each pixel point p , and A can refer to the area of the component to which the pixel point p belongs.

[0058] The weight coefficient w_p' can be expressed as Equation 2 depending on whether the pixel point p is included in the background or in the foreground of the component.

20 **[0059]** [Equation 2]

$$\mathbf{[0060]} \quad w_p' = \begin{cases} w_{av} & \text{if } p \in \text{background} \\ \tau_i \cdot w_{av} & \text{if } p \in C_i (i^{th} \text{ component}) \end{cases}$$

[0061] For example, when the pixel point p is included in the background, the weight coefficient w_p' can be defined as the mean coefficient w_{av} uniformly applied to the entire mask area, and

when the pixel point p is included in any component, the weight coefficient w_p' can be defined as $\tau_i \cdot w_{av}$.

[0062] Here, τ_i can refer to the component-wise weight, ensuring that the cumulative contribution of loss for each component is
5 assigned equally regardless of the component's area.

[0063] The component-wise weight τ_i can be defined as Equation 3.

[0064] [Equation 3]

[0065]
$$\tau_i = \frac{\frac{1}{K} \sum_j^K A_{C_j}}{A_{C_i}}$$

[0066] Here, K can refer to the number of components, A_{C_i} can
10 refer to the area of the i th component, and A_{C_j} can refer to the area of the j th component.

[0067] For example, assuming that there are two components (C_1 , C_2) in the image, and the area of component C_1 (A_{C_1}) is relatively larger than the area of component C_2 (A_{C_2}) ($A_{C_1} \gg A_{C_2}$), the weight
15 (τ_1) applied to component C_1 and the weight (τ_2) component C_2 can be calculated as follows:

[0068]
$$\tau_1 = \frac{\frac{1}{2} \sum_j^2 A_{C_j}}{A_{C_1}} = \frac{\frac{1}{2}(A_{C_1} + A_{C_2})}{A_{C_1}}, \quad \tau_2 = \frac{\frac{1}{2}(A_{C_1} + A_{C_2})}{A_{C_2}}$$

[0069] Equations 1 to 3 are applicable when processing the entire mask, however, the Component-Balance Focal (CBF) loss calculation
20 method can also be applied to point-based loss calculation, which calculates the loss at sampled points.

[0070] In point-based loss calculation, the loss (L_{CBF}) may be calculated based on the modified equations, Equations 4 to 6, derived from Equations 1 to 3.

[0071] [Equation 4]

$$[0072] \quad L_{CBF} = \sum_p^{N_{points}} w_p' \cdot l_{bce,p}$$

[0073] [Equation 5]

$$[0074] \quad w_p' = \begin{cases} w_{av} & \text{if } p \in \text{background} \\ \tau_i \cdot w_{av} & \text{if } p \in C_i (i^{th} \text{ component}) \end{cases}$$

5 [0075] [Equation 6]

$$[0076] \quad \tau_i = \frac{\frac{1}{K} \sum_j^K N_{C_j}}{N_{C_i}}$$

[0077] In Equations 4 to 6, N_{points} can refer to the number of sampled points, w_p' can refer to the weight coefficient, w_{av} can refer to the mean coefficient uniformly applied to the entire mask area, $l_{bce,p}$ can refer to the BCE loss for point p, K can refer to the number of components, N_{C_i} can refer to the number of sampled points for the i^{th} component, and N_{C_j} can refer to the number of sampled points for the j^{th} component.

[0078] Here, τ_i can refer to the component-wise weight, ensuring that the cumulative contribution of loss for each component is assigned equally regardless of the number of sampled points.

[0079] FIG. 2 is a flowchart illustrating an example of an instance segmentation model training method.

[0080] The stepwise operations depicted in FIG. 2 can be implemented by the processor 130 of the instance segmentation model training system 100 of FIG. 1.

[0081] With reference to FIG. 1 and FIG. 2, the processor 130

can receive image data at step S210 and apply the received image data to the instance segmentation model to detect an instance mask at step S220.

[0082] Sequentially, the processor 130 can compare the detected
5 instance mask with the GT instance mask at step S230 and calculate the CBF loss based on the loss calculation method at step S240.

[0083] At step S240, when configured to calculate the loss based on the entire mask, the processor 130 can compute the CBF loss based on Equations 1 to 3.

10 **[0084]** At step S240, when configured to calculate the loss based on sampled points, the processor 130 can compute the CBF loss based on Equations 4 to 6.

[0085] Afterwards, the processor 130 can update the instance segmentation model by incorporating the CBF loss at step S250.

15 **[0086]** Afterwards, the processor can may determine whether the predefined stopping condition is met at step S260.

[0087] If the predefined stopping condition is met (S260-Yes), the processor 130 can stop training for the instance segmentation model. If the predetermined stopping condition is not met (S260-
20 No), the processor 130 can perform steps S210 through S250 again. For example, the processor 130 can repeat steps 210 through 250 until the predefined stopping condition is met.

[0088] The stopping condition can be defined in a variety of ways depending on the application and/or model, and is not limited to
25 the present embodiment of the disclosure.

- [0089]** For example, the stopping condition can include when the difference (or loss) between the detected instance mask and the GT instance mask is less than a predefined threshold, when the rate of change of the difference (or loss) between the detection
5 instance mask and the GT instance mask is less than a predefined threshold, when the accuracy (or F1-score) of a performance evaluation metric is greater than a predefined threshold, when the instance segmentation model has been trained a predefined number of times, and so on.
- 10 **[0090]** Table 1 shows the results tested for the instance segmentation model trained using the implementations of features described in this application and the instance segmentation model trained according to conventional techniques.

[0091] [Table 1]

Model	Application of CBF loss	AP _{val}	ICAP _{val}	AP _{test}	ICAP _{test}
QueryInst	X	43.6	36.6	44.0	37.0
QueryInst	O	45.5	38.1	45.6	38.1
QueryInst (OCP ⁺)	X	44.1	37.2	44.9	37.8
QueryInst (OCP ⁺)	O	46.3	39.0	46.2	39.0
Mask2Former	X	55.3	47.1	55.3	46.9
Mask2Former	O	56.8	49.1	56.9	49.1

- 15 **[0092]** The tests were conducted based on the OCHumanEX benchmark,

an occluded human instance segmentation dataset.

[0093] It can be observed that the performance of the instance segmentation model applying the CBF loss according to implementations of the present disclosure is superior to that of other models. This performance improvement can be attributed to the enhancement in instance completeness due to the application of the CBF loss.

[0094] Therefore, it can be concluded that the instance segmentation model trained based on the CBF loss according to implementations of the present disclosure can outperform other existing models, demonstrating that the CBF loss is highly effective for implementing instance segmentation models.

[0095] FIG. 3 is a graph representing an example of the component-level recall relative to the ratio of component area to instance area.

[0096] With reference to FIG. 3, it can be seen that recall is lower for smaller components compared to larger components.

[0097] When the CBF is applied loss according to implementations of the features described in this application, the recall performance of the instance segmentation model can improve.

[0098] Notably, applying the CBF loss can significantly improve the recall performance for both large and small components compared to conventional instance segmentation models. This demonstrates that the CBF loss is highly effective in improving instance completeness while ensuring that it does not compromise

the segmentation performance for large components.

[0099] FIG. 4 is a diagram illustrating an example of a comparison between the training results of the instance segmentation model according to implementations of the present disclosure and the existing instance segmentation model with the ground truth (GT) instance mask.

[0100] In FIG. 4, (a) represents the GT instance mask, (b) represents the detected instance mask output by the conventional instance segmentation model, and (c) represents the detected instance mask output by the instance segmentation model according to implementations of the present disclosure.

[0101] With reference to FIG. 4, the conventional instance segmentation model misidentifies component_a of instance_A as a component of instance_B (b), while the instance segmentation model according to implementations of the present disclosure can correctly identify component_a as belonging to instance_A (c).

[0102] In some implementations, the processor 130 can be implemented as a hardware data processing device with a circuit having a physical structure to execute desired operations. For example, the desired operations can encompass codes or instructions included in the program.

[0103] For example, the hardware-implemented data processing device can encompass microprocessors, central processing units (CPUs), processor cores, multi-core processors, multiprocessors, application-specific integrated circuits (ASICs), and field

programmable gate arrays (FPGAs).

[0104] The processor 130 can be configured to store data related to algorithms or software instructions that control the operation of various components related to the features of the disclosure
5 in memory, and to execute the operations described herein using the data stored in the memory. Here, the memory and processor can be implemented as separate chips or integrated into a single chip.

[0105] For example, the memory can store various algorithms, data, and other information necessary for the operation of the processor
10 130. The memory can include volatile memory and/or non-volatile memory. The volatile memory can include dynamic random access memory (DRAM), static RAM (SRAM), synchronous DRAM (SDRAM), phase-change RAM (PRAM), magnetic RAM (MRAM), resistive RAM (RRAM), and ferroelectric RAM (FeRAM). The non-volatile memory
15 can include read-only memory (ROM), programmable ROM (PROM), electrically programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), and flash memory.

WHAT IS CLAIMED IS:

1. An instance segmentation model training method comprising:
applying image data to an instance segmentation model to
thereby detect an instance mask;

5 calculating a loss based on (i) the detected instance mask
and (ii) a ground truth (GT) instance mask corresponding to the
image data; and

 updating the instance segmentation model based on the
calculated loss,

10 wherein the loss is calculated by applying a component-wise
weight to a binary entropy loss calculation formula.

2. The method of claim 1, wherein the component-wise weight
is set based on a proportion of a component in the mask.

15

3. The method of claim 1 or claim 2, wherein the component-
wise weight is (i) not applied to an area corresponding to a
background and (ii) applied to an area corresponding to a
component of an instance.

20

4. The method of any one of claims 1 to 3, wherein the loss
is calculated based on an entire mask or sampled points.

5. The method of any one of claims 1 to 4, wherein the loss
25 is calculated based on an entire mask, using Equation 1:

[Equation 1]

$$L_{CBF} = \sum_p^A w'_p \cdot l_{bce,p},$$

where w'_p denotes a weight coefficient, $l_{bce,p}$ denotes a
 5 binary cross entropy (BCE) loss for each pixel point p , and A
 denotes an area of a component to which a pixel point p belongs.

6. The method of claim 5, wherein the weight coefficient w'_p
 is calculated using Equation 2 and Equation 3:

10 [Equation 2]

$$w'_p = \begin{cases} w_{av} & \text{if } p \in \text{background} \\ \tau_i \cdot w_{av} & \text{if } p \in C_i (i^{th} \text{ component}) \end{cases},$$

[Equation 3]

$$\tau_i = \frac{\frac{1}{K} \sum_j^K A_{C_j}}{A_{C_i}},$$

where w_{av} denotes a mean coefficient uniformly applied to
 15 an entire mask area, τ_i denotes the component-wise weight, K
 denotes a number of components, A_{C_i} denotes the area of the i^{th}
 component, and A_{C_j} denotes the area of the j^{th} component.

7. The method of any one of claims 1 to 6, wherein the loss
 20 is calculated based on sampled points, using Equation 4:

$$L_{CBF} = \sum_p^{N_{points}} w'_p \cdot l_{bce,p},$$

calculate a loss based on the detected instance mask
 and a GT instance mask corresponding to the image data, and
 update the instance segmentation model based on the
 calculated loss,
 5 wherein the loss is calculated by applying a component-wise
 weight to a binary entropy loss calculation formula.

10. The system of claim 9, wherein the component-wise weight
 is set based on a proportion of a component in the mask.

10

11. The system of claim 9 or claim 10, wherein the processor
 is configured to (i) not apply the component-wise weight to an
 area corresponding to a background and (ii) apply the component-
 wise weight to points of an area corresponding to a component of
 15 an instance.

12. The system of any one of claims 9 to 11, wherein the
 processor is configured to calculate the loss based on an entire
 mask or sampled points.

20

13. The system of any one of claims 9 to 12, wherein the
 processor is configured to calculate the loss based on an entire
 mask, using Equation 1:

$$L_{CBF} = \sum_p^A w'_p \cdot l_{bce,p}$$

where w'_p denotes a weight coefficient, $l_{bce,p}$ denotes a binary cross entropy (BCE) loss for each pixel point p , and A denotes an area of a component to which a pixel point p belongs.

5

14. The system of claim 13, wherein the processor is configured to calculate the weight coefficient w'_p using Equation 2 and Equation 3:

[Equation 2]

$$10 \quad w'_p = \begin{cases} w_{av} & \text{if } p \in \text{background} \\ \tau_i \cdot w_{av} & \text{if } p \in C_i (i^{\text{th}} \text{ component}) \end{cases},$$

[Equation 3]

$$\tau_i = \frac{\frac{1}{K} \sum_j^K A_{C_j}}{A_{C_i}},$$

where w_{av} denotes a mean coefficient uniformly applied to an entire mask area, τ_i denotes the component-wise weight, K denotes a number of components, A_{C_i} denotes the area of the i^{th} component, and A_{C_j} denotes the area of the j^{th} component.

15

15. The system of any one of claims 9 to 14, wherein the processor is configured to calculate the loss based on sampled points, using Equation 4:

20

$$L_{CBF} = \sum_p^{N_{\text{points}}} w'_p \cdot l_{bce,p},$$

where N_{points} denotes a number of sampled points, w'_p denotes a weight coefficient, $l_{bce,p}$ denotes a binary cross entropy (BCE) loss for each point p .

5 16. The system of claim 15, wherein the processor is configured to the weight coefficient w'_p using Equation 5 and Equation 6:

[Equation 5]

$$w'_p = \begin{cases} w_{av} & \text{if } p \in \text{background} \\ \tau_i \cdot w_{av} & \text{if } p \in C_i (i^{th} \text{ component}) \end{cases},$$

10 [Equation 6]

$$\tau_i = \frac{\frac{1}{K} \sum_j^K N_{C_j}}{N_{C_i}},$$

where w_{av} denotes a mean coefficient uniformly applied to an entire mask area, τ_i denotes the component-wise weight, K denotes a number of components, N_{C_i} denotes a number of sampled points for the i^{th} component, and N_{C_j} denotes a number of sampled points for the j^{th} component.