DOI: https://doi.org/10.15276/hait.08.2025.4
UDC 004:83

Improved segmentation model to identify object instances based on textual prompts

Andrii R. Kovtunenko[1)]
ORCID: https://orcid.org/0009-0004-9072-7779; andrii.kovtunenko@nure.ua. Scopus Author ID: 58362751200
Sergii V. Mashtalir[1)]
ORCID: https://orcid.org/0000-0002-0917-6622; sergii.mashtalir@nure.ua. Scopus Author ID: 36183980100
[1)] Kharkiv National University of Radio Electronic, 14, Nauky Ave. Kharkiv, 61166, Ukraine

ABSTRACT

The rapidly increasing amount of multimedia information requires significant methods development for its rapid processing. In this case, one of the areas of processing is preliminary analysis with the images characteristic features detection to reduce the information required for subsequent tasks. One of the types for an information reduction is image segmentation. In this case, the general task of image segmentation is often reduced to the task of object segmentation is a fundamental task in computer vision, requiring accurate pixel-by-pixel object delineation and scene understanding. With the development of natural language processing techniques, many approaches have been successfully adapted to computer vision tasks, allowing for more intuitive descriptions of scenes using natural language. Unlike traditional models limited to a fixed set of classes, natural language processing-based approaches allow searching for objects based on attributes, expanding their applicability. While existing object segmentation methods are typically categorized into one-stage and two-stage methods – depending on speed and accuracy - there remains a gap in developing models that can effectively identify and segment objects based on textual prompts. To address this, we propose an open-set instance segmentation model capable of detecting and segmenting objects from prompts. Our approach builds upon CLIPSeg, integrating architectural modifications from Panoptic-DeepLab and PRN (Panoptic Refinement Network) to predict object centers and pixel-wise distances to boundaries. A post-processing phase refines segmentation results to improve object separation. The proposed architecture is trained on large vocabulary instance segmentation and PhraseCut datasets and evaluated using the mean Dice score against state-of-the-art open-set segmentation models. Experimental results show that although our model achieves the highest inference rate among open-set methods while maintaining FastSAM-level segmentation quality, post-processing remains a limiting factor. This suggests that future improvements should be aimed at eliminating the post-processing process itself or improving its algorithm, which could lead to more efficient segmentation.

Keywords: Deep learning; image segmentation; convolution neural networks; transformers; contrastive language-image pretraining; open-set segmentation

For citation: **Kovtunenko A. R., Mashtalir S. V. "Improved segmentation model to identify object instances based on textual prompts". *Herald of Advanced Information Technology*. 2025; Vol. 8 No. 1: 54–66. DOI: https://doi.org/10.15276/hait.08.2025.4**

INTRODUCTION

One of the fundamental functions of computer vision is to understand and interpret the surrounding space – images and videos. The reference approach for the solution has not been found yet, and the search process continues.

Among the approaches to understanding the surrounding space, segmentation should be highlighted. The process of segmentation is the division of data or images into logical and interrelated parts that represent objects, areas, or categories. The main purpose of segmentation is to simplify or transform the data into a more understandable and easier to analyze form, highlighting only the most relevant objects and details. Among the various types of segmentation, it is worth highlighting instance segmentation.

Instance segmentation is a computer vision task that involves semantic segmentation and the extraction of the boundaries of each object after it. Unlike semantic segmentation, which assigns a class label to each pixel without distinguishing individual objects, instance segmentation treats each occurrence of an object class as a separate entity. This provides a more detailed understanding of visual scenes. This approach is indispensable in scenarios where understanding the relationships, separation of objects within the same class, and their interactions is crucial. For example, in autonomous driving, accurate separation of individual pedestrians and cars is necessary for safe driving; in augmented reality, understanding individual objects in a scene; in robotics, identifying object instances and their boundaries for interaction with them, and so on.

1. RELATED WORKS

Approaches for instance segmentation can be categorized as follows: one-stage – aimed at predicting objects and their masks in one step, without using additional methods to predict

regions of interest or find objects: YOLACT [1], SOLO [2], PolarMask [3], MEInst [4], CenterMask [5].

This approach gains in speed but may lose in accuracy: two-stage – the task is divided into two separate phases: region-of-interest detection and then segmentation to refine a pixel-level mask for each object instance. These may include: Mask R-CNN [6], RefineMask [7], Mask Transfiner [8], TensorMask [9], Polytransform [10], BCNet [11]. In contrast, these methods are more accurate but require more computational resources.

In turn, they can also be further divided into:

– Top-down (detection first) – the target objects are first detected, and then their segmentation masks are refined.

– Bottom-up (segmentation first) – relevant individual features are first identified and then grouped into instances or segments belonging to the same object.

In other words, in the top-down approach, the goal is to first find objects and then assign unique identifiers to them and refine the boundaries if the objects were searched through object detection. In the bottom-up approach, the goal is to first identify the features that make it clear that this is the object of interest and then combine these features into different objects. This approach is well illustrated by finding features for each pixel and then clustering them.

It is also possible to combine these methods and approaches to solve the problem at hand. As a rule, bottom-up methods lag in accuracy compared to top-down methods, especially on a data set of complex shapes and a large variety of them. The top-down ones, on the other hand, have problems with small-sized objects and may produce multiple overlapping masks, which additionally require post-processing.

The above-presented methods were trained on a fixed data set and require additional training when changing the format or type of input data, for example, adding a new class to be found. In order for the model to be more robust to changes, it should be trained on a large variant dataset, which makes the training process long and costly.

Zero-shot or few-shot learning approaches are designed to solve this problem.

Zero-shot learning (ZSL) is an approach in machine learning in which a model can solve problems related to categories or tasks for which it has not been explicitly trained. This means that the model can recognize objects that belonged to new classes that did not occur in the training data, but based on additional information, the model can understand this.

The few-shot learning (FSL) approach, like ZSL, aims at providing the ability of the model to recognize previously unknown objects by training on a small amount of data.

The idea of allocating unique semantic features is not new, and the appearance of transformer architectures, in particular the CLIP model (Contrastive Language-Image Pretraining) [12], allowed to approach this and made it possible to combine a textual description of features, which is more understandable to a human, with the characteristics of objects in images.

A true revolution in zero-shot, for the instance segmentation task, was made by the SAM (Segment Anything Model) [13] model. Its goal is to provide a universal solution for segmenting objects in images, regardless of their category. This is made possible by training the model on the huge SA-1B [13] dataset. Initially, the image is converted into embeddings using a modified ViT (Visual Transformer) model that takes a 1024x1024 image as input. Then either a text description of the searched objects or their location is added to previous data, and masks of the searched objects are obtained using a lightweight decoder.

Also, SAM varieties have been developed to solve the problems of inaccurate object boundary extraction and execution speed: HQ-SAM (High-Quality SAM) [14], Grounded-SAM [15], and Fast-SAM [16]. At the moment, there is no SAM implementation where objects can be specified by text prompt. In the two-stage Grounded-SAM method, objects are first selected from a textual description using GroundingDINO [17], whose locations are then passed to SAM to obtain masks.

2. PROBLEM STATEMENT

In this work, we propose an improved InstanceCLIPSeg method, which is based on the CLIPSeg semantic segmentation model. The architecture of CLIPSeg is modified by replacing the semantic decoder with two decoders that will find separately the centers of objects and the distance of an object pixel to each of its four boundaries. The obtained results are merged into instances by postprocessing.

To date, there is no one-stage open-set model for solving the instance segmentation problem. The aim of this paper is to create a one-stage open-set model for solving the instance segmentation problem using textual descriptions of objects or

regions to be selected, based on the CLIPSeg [18] method by improving it – changing the architecture.

To achieve this goal, it is necessary to develop a modification of the CLIPSeg semantic segmentation model and conduct an experimental analysis of its effectiveness relative to other segmentation models.

The subject of this research is instance segmentation methods and algorithms that use textual descriptions of objects or regions in images to select each instance of an object to implement a zero-show or few-shot approach.

The object of the study is the process of finding regions in images from textual descriptions and then combining them into object instances.

Limitations – a text query for object retrieval should describe objects belonging to the same searched category and should not have contradictory features. If multiple semantics are to be found, there should be multiple queries too.

3. PROPOSED METHOD

Like the original model, our model consists of a CLIP encoder (ViT-B/16), which was adapted for 352x352 resolutions, a prompt encoder, and two decoders for each of the heads – centers head, offset head. Centers head – predicts the center of mass of objects encoded by Gaussian. Offset head – predicts the distance of each pixel belonging to an object to the boundaries of this object (Fig. 1).

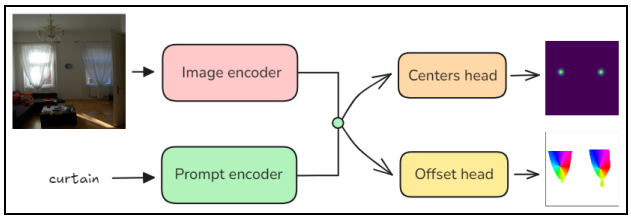


Fig. 1. **InstanceCLIPSeg architecture**
Source: **compiled by the authors**

In CLIPSeg, TransposedConvolution was used to increase the dimensionality of the obtained features after TransformerEncoderLayer to restore the original image size. This approach creates artifacts [19] and does not allow obtaining consistent pixel values in the regions, which is exactly what is required in our approach to accurately predict the center and distance of a pixel to the boundaries (Fig. 2).

Therefore, in the centers head, sequential TransposedConvolution was replaced by four consecutive blocks to restore the size from 64 to 352 pixels. The block consists of two sub-blocks: dimensionality increase and refinement. The dimensionality increase block consists of TransposedConvolution with kernel size four, stride two, padding one; batch normalization; ReLU activation function. The refinement block consists of convolution with kernel size three, stride one, padding one; batch normalization; ReLU activation function.

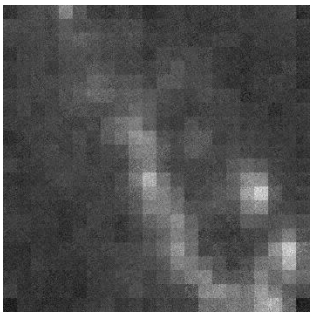


Fig. 2. **Artifacts after TransposedConvolution**
Source: **compiled by the authors**

The offset head requires more context and smooth output, so a PixelShuffle layer was used for it in a sub-block to increase dimensionality. This sub-block consists of a convolution with kernel size three, stride one and padding one; Normalization Batch, ReLU activation functions; PixelShuffle with upscale factor two. The refinement block is the same as the block from the centers head. The block amount in the offset head is also four.

The approach that the model should predict object centers and pixels belonging to the object as the distance to some point inside the object or to the boundaries was taken from Panoptic-DeepLab [20]. But, since our model uses the ReLU activation function everywhere, the original approach, where the offset head predicts the distance of an object pixel to its center, which can be negative, is not suitable for us. During experiments, it was found that negative values in the last step are difficult to predict, so this approach was replaced by predicting the distances to each of the four object boundaries as proposed in PRN [21]. So far, we have completely abandoned the additional prediction of the background and the distance from the center to the pixels because the objects searched by text description will belong to the same category, the background can be found by inverting the resulting segmentation mask.

4. POST-PROCESSING

After obtaining the prediction of object centers position and distance to boundaries, post-processing is performed. Threshold and Non-Maximum Suppression (NMS) are applied to the centers' heatmap to obtain the most probable centers and to reduce noise in the image. After that, coordinate maps are constructed in which pixels are numbered

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

from zero to the image size from each of the boundaries, from which the resulting offset values are then subtracted. After this step, each object will have equal, with some error, values and can be selected. This is done by clustering near the obtained centers by distance to them and pixel values near them. This approach allows selecting overlapping objects if the overlap is correctly predicted and does not require a lot of computational resources. In turn, this approach is a drawback and requires improvement, as it is highly dependent on accurate offset head prediction, which reduces the accuracy.

5. TRAINING

The model was fine-tuned using the Adam optimizer at batch size 64. The original CLIPSeg weights were loaded into the rest of the model and were pre-trained along with the new layers. The centers head loss function was a weighted Mean Square Error (MSE) where the center was marked with a factor of 10 and background one. The offset head loss function was a weighted L1. The model was trained for 20 epochs using SequentialLR scheduler, where LinearLR was used for warmup and then ExponentialLR from $1*10^{-3}$ to $1*10^{-4}$. The datasets used were LVIS [22] and PhraseCut [23], which will be discussed further below. Input data were augmented with Rotate, Flip RandomScale between one and two.

6. EXPERIMENTS

Two datasets, LVIS and PhraseCut, were selected for training. LVIS has the advantage of containing many labelled objects per image, while PhraseCut is more focused on understanding the context of objects in images. Below are the statistics of image sizes and object sizes by dataset (Fig. 3-7).

For the CLIP image encoder and CLIP prompt encoder, the following experiment was performed. We took the CIFAR100 dataset [24] for 100 categories and added a 101st category with an object name that was not in the original dataset. Text embedding was counted for each category. Image embeddings were calculated as follows. A square image of the object was taken and placed in the center on a uniform background with the object zoomed in 1px increments. The experiment was conducted with the original CLIP encoder 224x224px and with the improved one 352x352px. After that, the cosine distance between image embeddings and text embeddings was calculated, and the distances between the classes from CIFAR100 and the image were taken randomly. The results are presented in the Fig. 8. Red shows classes that are not in the image, blue shows the class we are looking for.

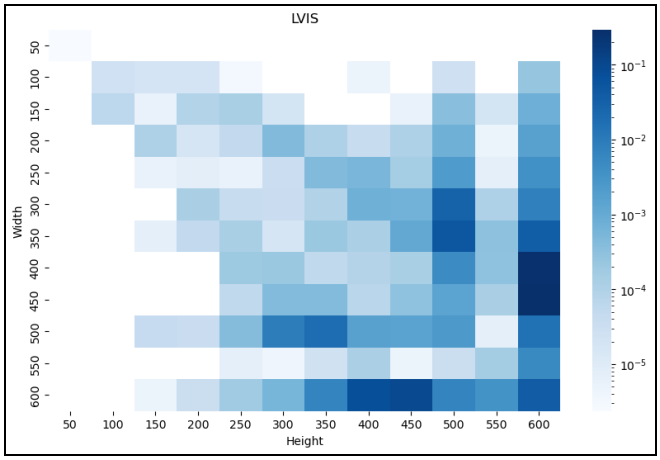


Fig. 3. **Statistics of image sizes in LVIS**
Source: **compiled by the authors**

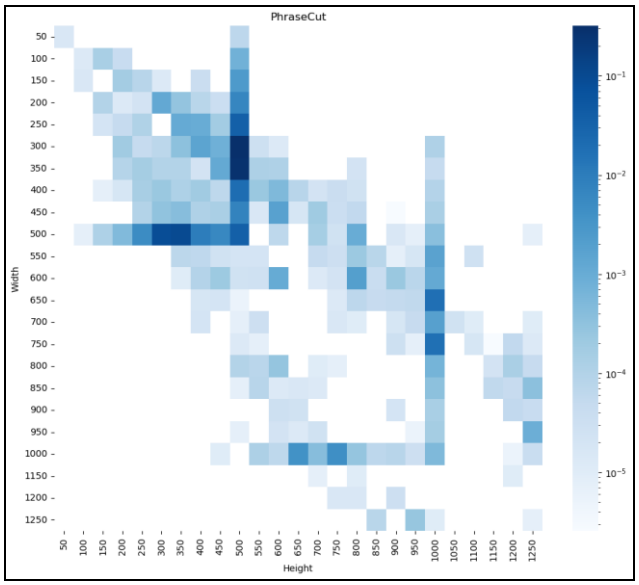


Fig. 4. **Statistics of image sizes in PhraseCut**
Source: **compiled by the authors**

As can be seen from the experiment, increasing the resolution to 352x352px allowed better class separation, but it is still difficult to distinguish small objects up to 10px for the model. Therefore, based on the above-presented statistics on datasets, we decided to train the model only on objects whose area after resizing is 100px or more.

We compared our model with existing State-of-the-art models in terms of a number of parameters, speed in time and frame per second (fps), and the ability of the models to find objects from a textual description (Table 1 and Table 2).

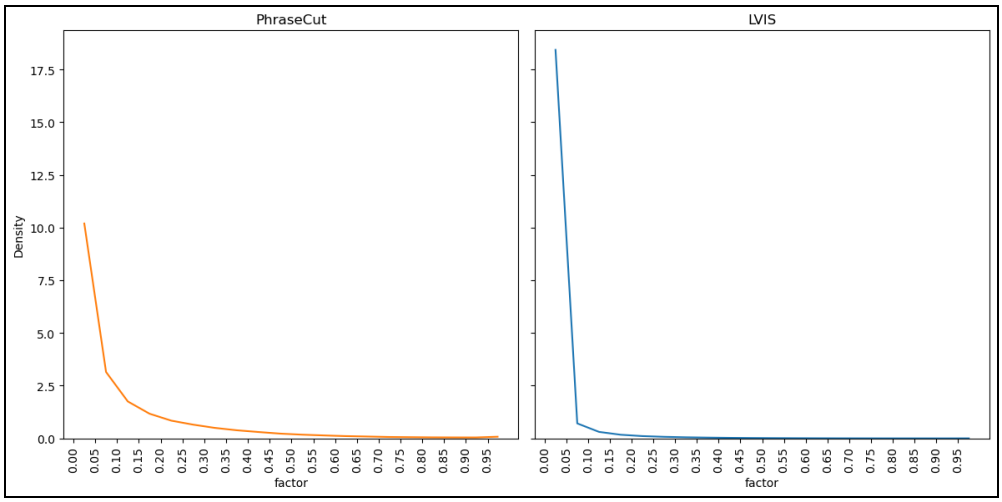


Fig. 5. **A statistic of the ratio of the area value of an object in an image to the area of the whole image**
Source: **compiled by the authors**

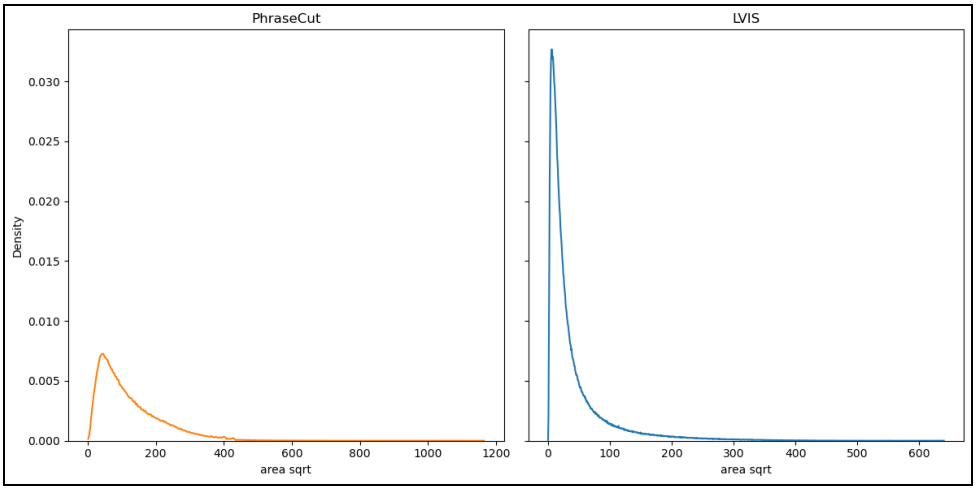


Fig. 6. **Statistics of the area of the selected area of the object in the form of a square in pixels on the original image**
Source: **compiled by the authors**

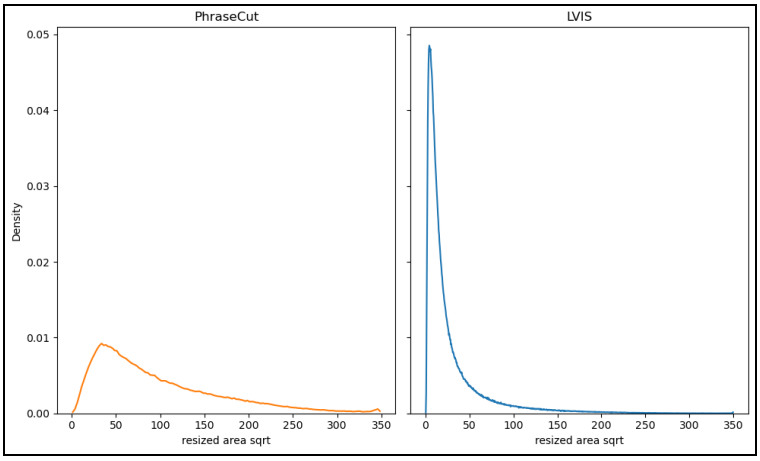


Fig. 7. **Statistics of the area of the selected area of the object in the form of a square in pixels on the image 350x350px**
Source: **compiled by the authors**

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

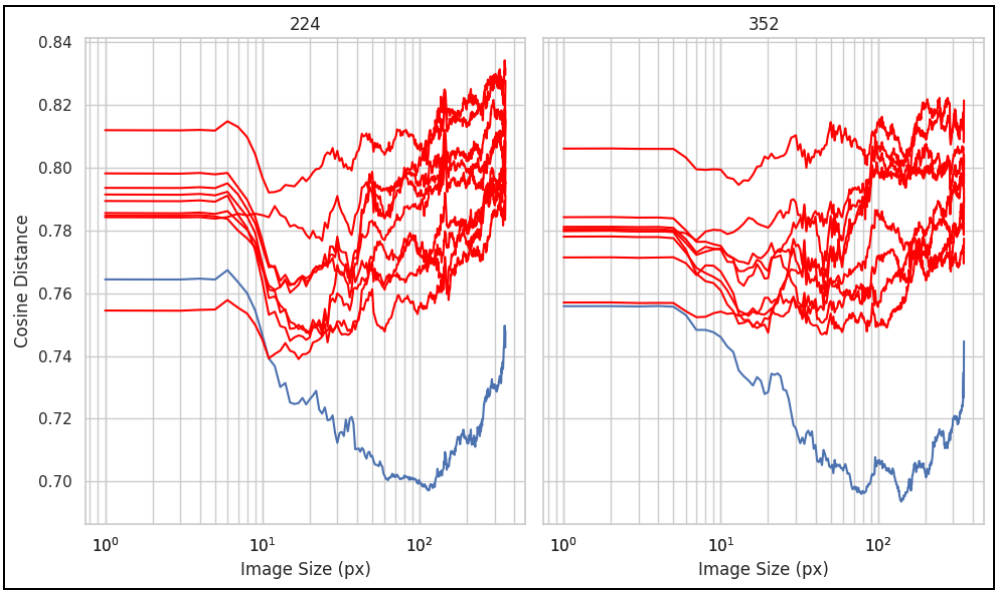


Fig. 8. **Cosine distances of text embeddings to image embeddings as a function of object size in the image**
Source: **compiled by the authors**

Table 1. **Model comparison**

Model	Parameters	Time	Fps
HQ-SAM ViT-B	358M	0.11s	9.07
SAM ViT-B	362.1M	0.101s	9.86
Grounded SAM	358M + 232.3M	0.269s	3.71
FastSAM	72M	0.04s	25
FastSAM text	72M + 151M	1.1s	0.91
InstanceCLIPSeg	152.2M	0.045s	21.9

Source: **compiled by the authors**

Table 2. **Model comparison with mean Dice score**

Model	Mean Dice Score
HQ-SAM ViT-B	0.46506062
SAM ViT-B	0.4906734
FastSAM	0.21242003
InstanceCLIPSeg Centers	0.23045772
InstanceCLIPSeg Offsets	0.20319612

Source: **compiled by the authors**

We compared the speed of the models as it is from their GitHub repositories [25, 26], [27, 28] on a GTX 3090 Ti graphics card.

We measured the quality of finding objects using mean Dice coefficient (1) for all found objects on the PhraseCut test dataset.

Finding objects was checked only with a text prompt.

$$DS = \frac{2|X \cap Y|}{|X| + |Y|}, \quad (1)$$

where X is pixel sets greater than zero of the output image, and Y is pixel sets on the ground truth image, $|\circ|$ is a cardinal number.

In the selected models, text search is realized only by using third-party open-set detectors. We used GroundingDINO for SAM and HQ-SAM, as suggested in the repositories. For FastSAM, the detector is YOLOv8 [29], but for text search, they also use the CLIP ViT-B/32 model, which significantly increases the number of parameters and slows down the execution speed. For our model, we compared each of the decoders separately. As can be seen, centers are located more accurately than offsets. Our model outperforms all other compared models in terms of speed for retrieval using textual description. In terms of quality, it is comparable to FastSAM and also has the advantages of a one-stage approach.

7. RESULTS AND DISCUSSIONS

This section shows the intermediate results of the model on instance segmentation task by text description and the final result of splitting into instances (Fig. 9 – Fig. 15). Figures descriptions are shown in Table 3.

In Fig. 9, the model detected cars in the distance using the centers head, but the offset head failed to estimate the distances, resulting in only one car being identified in the final output.

A similar situation is observed in Fig. 10 with people, where the model struggles to distinguish overlapping objects.

In Fig. 11, we attempted to find small objects. As we can see, only part of the objects was detected, and one bush was split into two due to the centers head predicting two centers. This issue can be corrected by adjusting the NMS parameters, but doing so would compromise the model's general applicability for detecting arbitrary objects, as specific parameters would need to be fine-tuned for each case.

In Fig. 12, parts of the elephants' boundaries were incorrectly assigned. This happened because the offset head produced only two instances, and we attempted to mitigate this issue during post-processing by considering distances to the centers. Although the third elephant, which is partially obscured by a baby elephant, was poorly predicted by the centers head, post-processing was able to partially highlight it.

In Fig. 14, overlapping objects again caused difficulties. Since the prompt did not specify which monitors should be detected, the model also identified the projector screen. However, the turned-off monitors were not detected by the offset head, nor were the distant objects.

In Fig. 15, the model produced an acceptable result based on the provided queries.

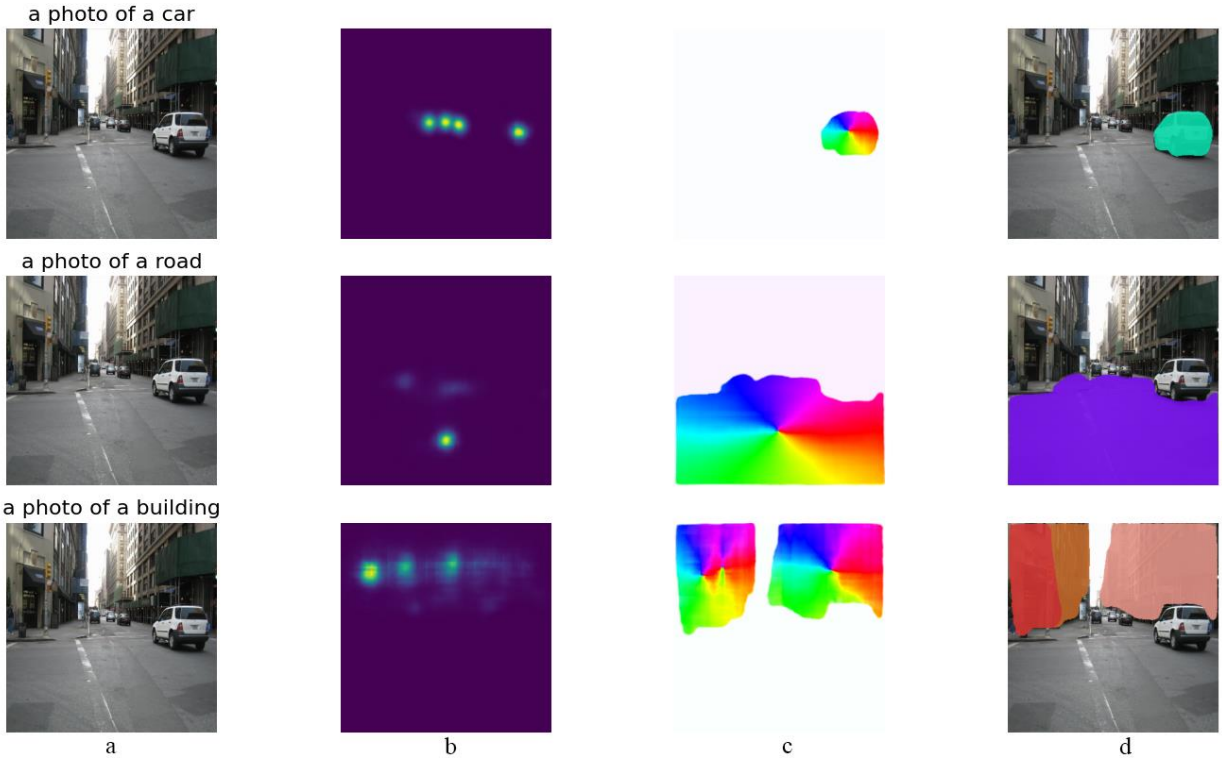


Fig. 9. **Examples of model execution followed by clustering of city street image (Letters description shown in Table 3)**

Source: **compiled by the authors**

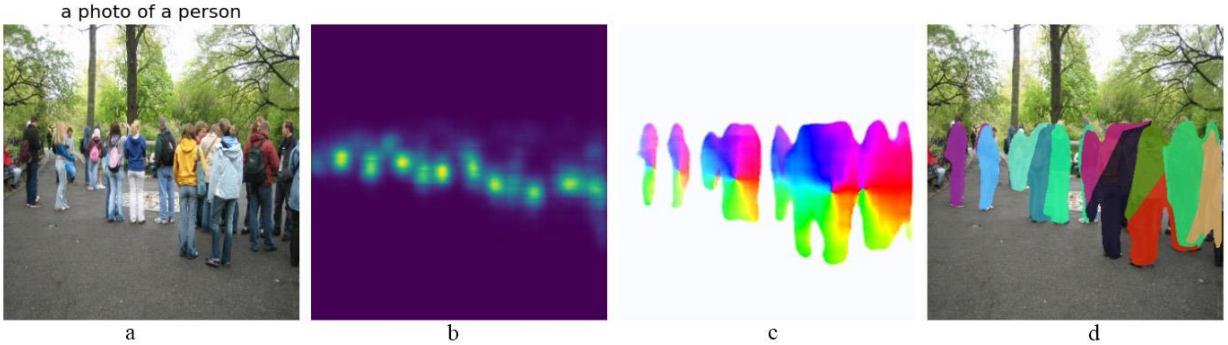


Fig. 10. **Examples of model execution followed by clustering for persons image (Letters description shown in Table 3)**

Source: **compiled by the authors**

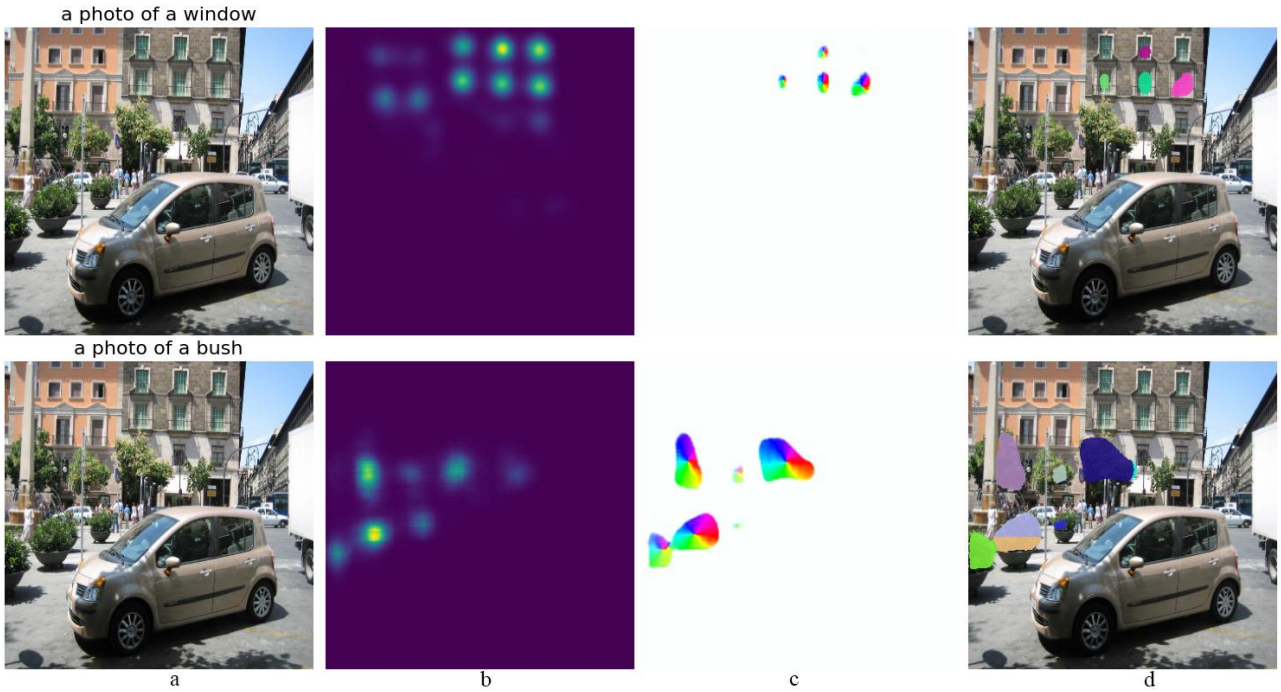


Fig. 11. **Examples of model execution followed by clustering of outdoor image (Letters description shown in Table 3)**
Source: **compiled by the authors**

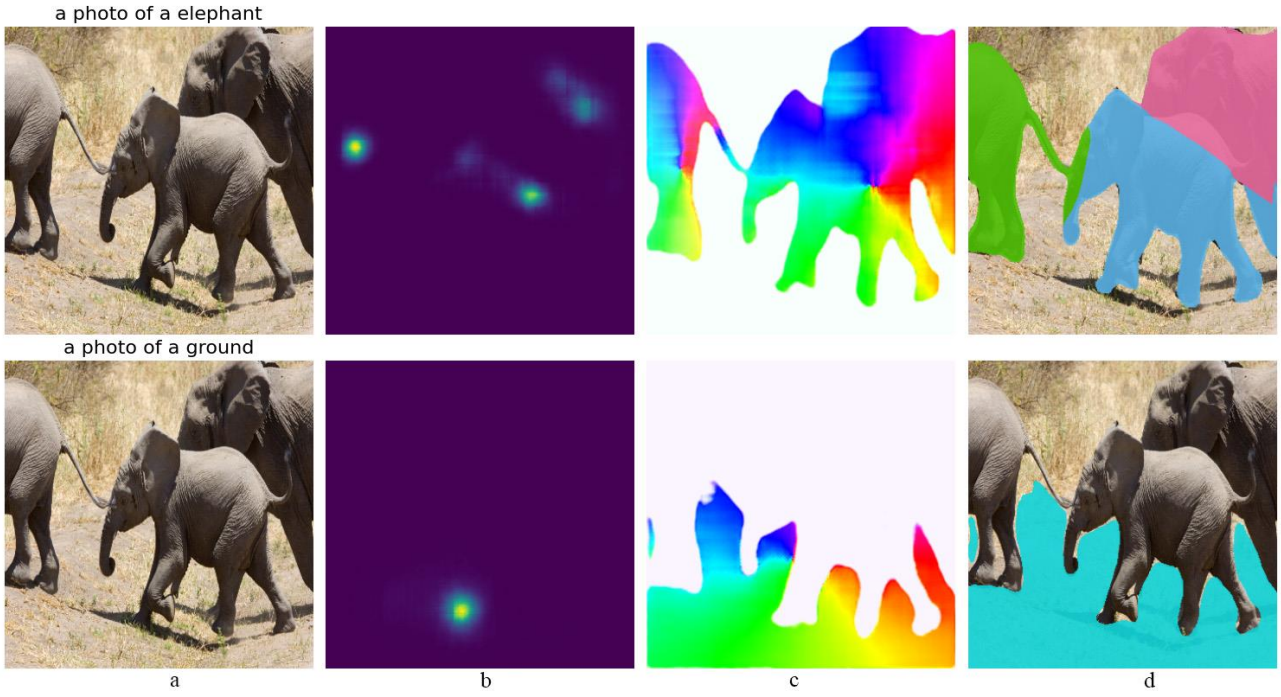


Fig. 12. **Examples of model execution followed by clustering of elephants image (Letters description shown in Table 3)**
Source: **compiled by the authors**

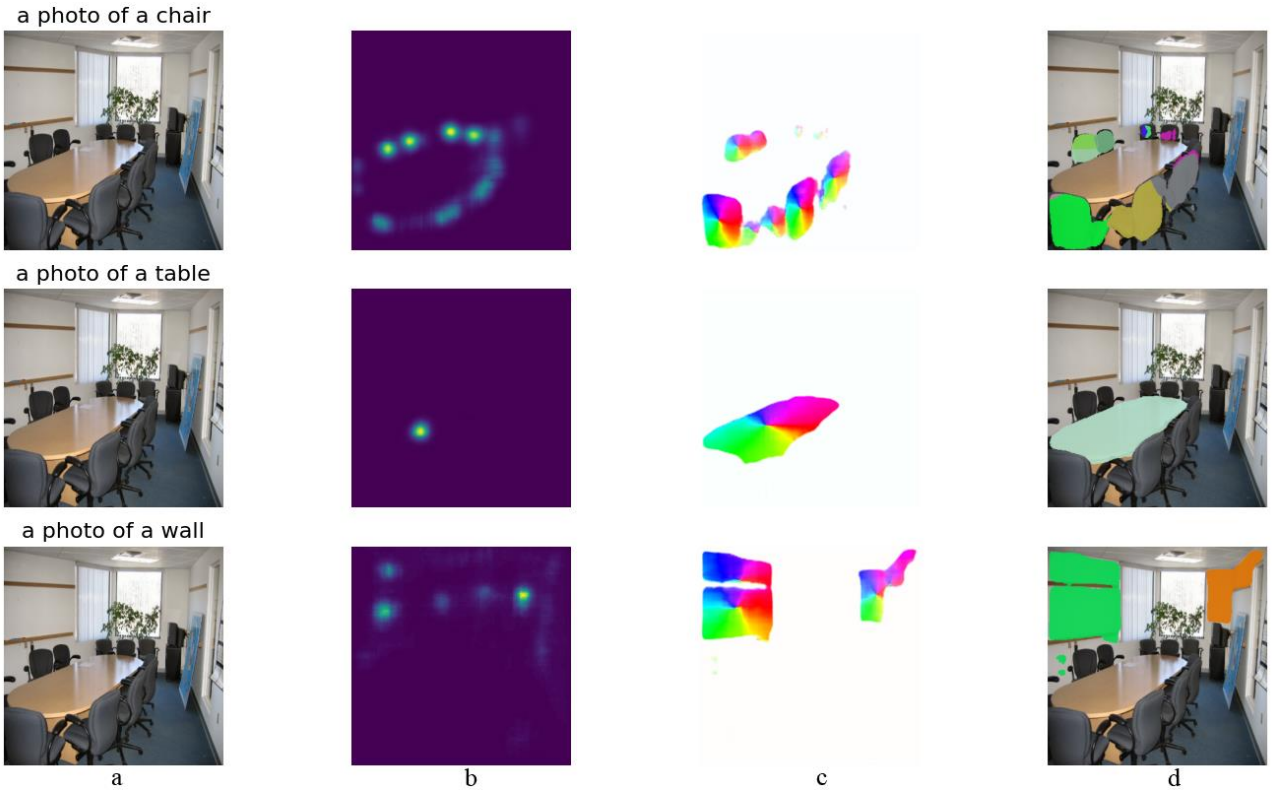


Fig. 13. **Examples of model execution followed by clustering of conference room image (Letters description shown in Table 3)**
Source: **compiled by the authors**

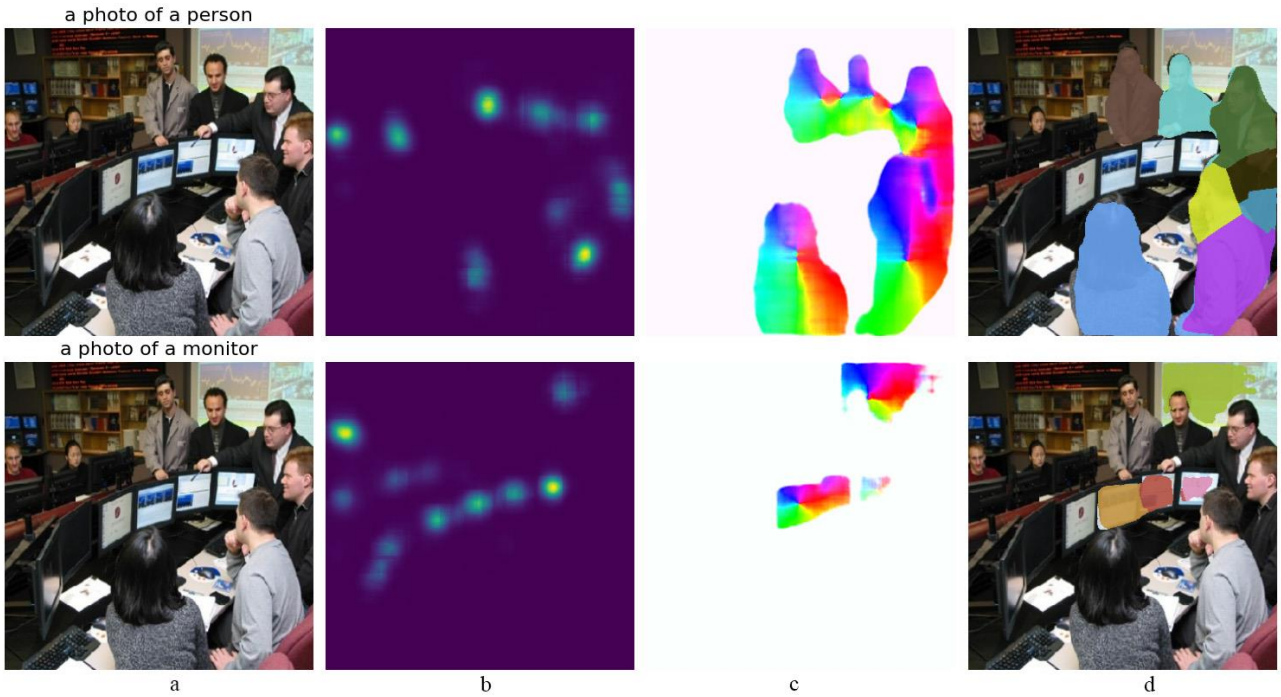


Fig. 14. **Examples of model execution followed by clustering of meeting image (Letters description shown in Table 3)**
Source: **compiled by the authors**

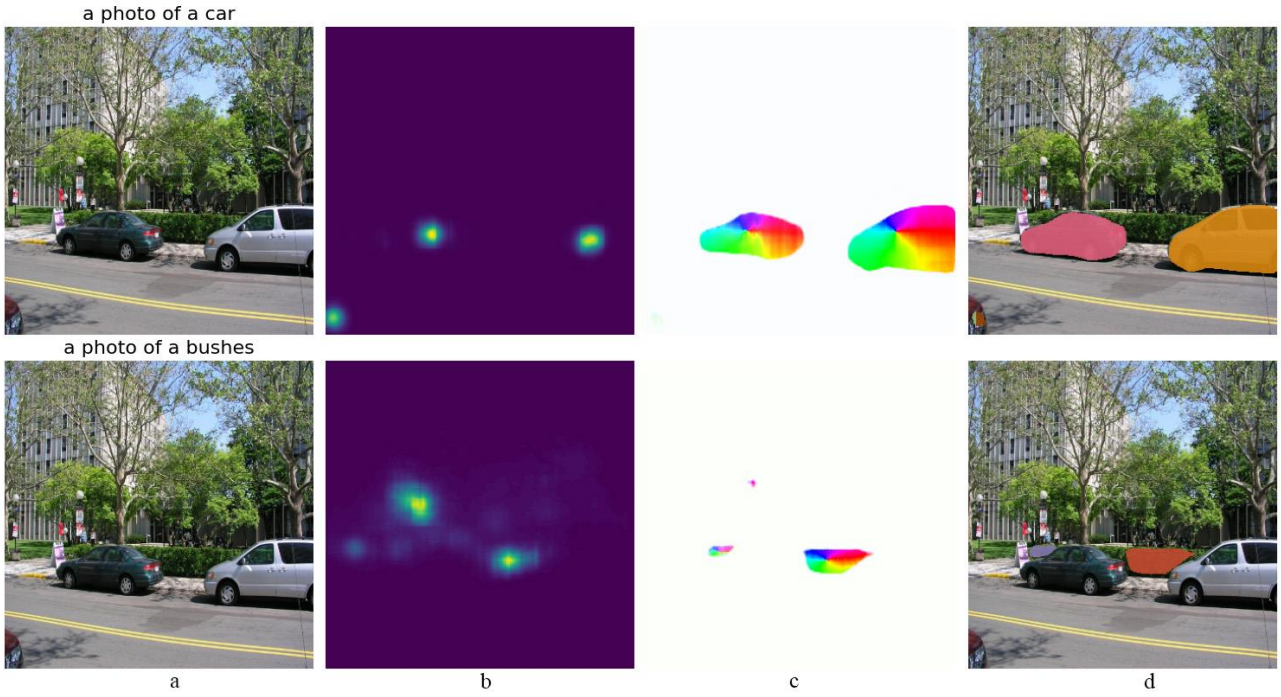


Fig. 15. **Examples of model execution followed by clustering of a cars image (Letters description shown in Table 3)**
Source: **compiled by the authors**

Table 3. **Figures description**

Letter	Description
a	The original image fed to the model as input and the text description of the objects being searched
b	The output of the centers head
c	The output of offset head
d	The result of splitting into instances after post-processing

Source: **compiled by the authors**

CONCLUSIONS AND PROSPECTS OF FURTHER RESEARCH

This paper presents an improved version of the CLIPSeg model, InstanceCLIPSeg, designed to solve the instance segmentation problem. Improvements include modification of the architecture and integration of techniques borrowed from other models. InstanceCLIPSeg can find objects from their textual description, proving its zero-shot learning ability.

We analyzed the training datasets and optimized the training parameters to improve the segmentation quality. In addition, we developed a post-processing method that uses the predicted object centers and distances to their boundaries to separate them. The final model contains 152.2M parameters and achieves a speed of 20 FPS on 352×352 images using an RTX 3090 Ti GPU.

InstanceCLIPSeg was compared with existing state-of-the-art open-set instance segmentation models by the mean Dice score, number of parameters, and speed of operation. Experimental results showed that the proposed approach is comparable to state-of-the-art solutions and can perform the task efficiently.

Despite the achieved results, post-processing remains a bottleneck of the model, creating additional computational costs and potential segmentation errors. Also, the offset head with current architecture shows worse finding results than the centers head. Future research should focus on improving the model architecture or changing the representation of objects in the offset head with the post-processing algorithm and to improve prediction accuracy. Additional promising research directions include adapting the model to higher image resolutions, optimizing it for mobile devices, and exploring integration with transform architectures for more accurate object separation in complex scenes.

REFERENCES

1. Bolya, D., Zhou, C., Xiao, F. & Lee, Y. J. "Yolact: Real-time instance segmentation". *In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019. p. 9157–9166, https://www.scopus.com/authid/detail.uri?authorId=57215774696.
DOI: https://doi.org/10.1109/iccv.2019.00925.

2. Wang, X., Kong, T., Shen, C., Jiang, Y. & Li, L. "SOLO: Segmenting objects by locations". *In Computer Vision–ECCV 2020: 16th European Conference*, Glasgow, UK, August 23-28, 2020. DOI: https://doi.org/10.1007/978-3-030-58523-5_38.

3. Xie, E., Sun, P., Song, X., et al. "Polarmask: Single shot instance segmentation with polar representation". *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020. p. 12193–12202. DOI: https://doi.org/10.1109/cvpr42600.2020.01221.

4. Zhang, R., Tian, Z., Shen, C., You, M. & Yan, Y. "Mask encoding for single shot instance segmentation". *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020. p. 10226–10235. DOI: https://doi.org/10.1109/cvpr42600.2020.01024.

5. Lee, Y. & Park, J. "Centermask: Real-time anchor-free instance segmentation". *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020. p. 13906–13915. DOI: https://doi.org/10.1109/cvpr42600.2020.01392.

6. He, K., Gkioxari, G., Dollár, P. & Girshick, R. "Mask r-cnn". *In Proceedings of the IEEE international conference on computer vision*. 2017. p. 2961–2969. DOI: https://doi.org/10.1109/iccv.2017.322.

7. Zhang, G., Lu, X., Tan, J., Li, J., Zhang, Z., Li, Q. & Hu, X. "Refinemask: Towards high-quality instance segmentation with fine-grained features". *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021. p. 6861–6869. DOI: https://doi.org/10.1109/cvpr46437.2021.00679.

8. Ke, L., Danelljan, M., Li, X., Tai, Y. W., Tang, C. K. & Yu, F. "Mask transfiner for high-quality instance segmentation". *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022. p. 4412–4421. DOI: https://doi.org/10.1109/cvpr52688.2022.00437.

9. Chen, X., Girshick, R., He, K. & Dollár, P. "Tensormask: A foundation for dense object segmentation". *In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019. p. 2061–2069. DOI: https://doi.org/10.1109/iccv.2019.00215.

10. Liang, J., Homayounfar, N., Ma, W. C., Xiong, Y., Hu, R. & Urtasun, R. "Polytransform: Deep polygon transformer for instance segmentation". *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020. p. 9131–9140. DOI: https://doi.org/10.1109/cvpr42600.2020.00915.

11. Ke, L., Tai, Y. W. & Tang, C. K. "Deep occlusion-aware instance segmentation with overlapping bilayers". *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. p. 4019–4028. DOI: https://doi.org/10.1109/cvpr46437.2021.00401.

12. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S. & Sutskever, I. "Learning transferable visual models from natural language supervision". *In International Conference on Machine Learning*, 2021. p. 8748–8763, https://www.scopus.com/authid/detail.uri?authorId=24831264500. DOI: https://doi.org/10.48550/arXiv.2103.00020.

13. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L. & Girshick, R. "Segment anything". *In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023. p. 4015–4026, https://www.scopus.com/authid/detail.uri?authorId=35179333300.
DOI: https://doi.org/10.1109/iccv51070.2023.00371.

14. Ke, L., Ye, M., Danelljan, M., Tai, Y. W., Tang, C. K. & Yu, F. "Segment anything in high quality". *Advances in Neural Information Processing Systems*. 2024. p. 36. DOI: https://doi.org/10.48550/arXiv.2306.01567.

15. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H. & Zhang, L. "Grounded sam: Assembling open-world models for diverse visual tasks". *arXiv preprint*. 2024. DOI: https://doi.org/10.48550/arXiv.2401.14159.

16. Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M. & Wang, J. "Fast segment anything". *arXiv preprint*. 2023. DOI: https://doi.org/10.48550/arXiv.2306.12156

17. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J. & Zhang, L. "Grounding dino: Marrying dino with grounded pre-training for open-set object detection". *In European Conference on Computer Vision*, 2025. p. 38–55. *Springer. Cham*. DOI: https://doi.org/10.1007/978-3-031-72970-6_3.

18. Lüddecke, T. & Ecker, A. “Image segmentation using text and image prompts”. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2022. p. 7086–7096. DOI: https://doi.org/10.1109/cvpr52688.2022.00695.

19. Odena, A., Dumoulin, V. & Olah, C. “Deconvolution and checkerboard artifacts”. *Distill*, 2016; 1(10): e3. DOI: https://doi.org/10.23915/distill.00003.

20. Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H. & Chen, L. C. “Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation”. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2020. p. 12475–12485, https://www.scopus.com/authid/detail.uri?authorId=35594050800.
DOI: https://doi.org/10.1109/cvpr42600.2020.01249.

21. Sun, B., Kuen, J., Lin, Z., Mordohai, P. & Chen, S. “PRN: Panoptic Refinement Network”. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023. p. 3963–3973. DOI: https://doi.org/10.1109/wacv56688.2023.00395.

22. Gupta, A., Dollar, P. & Girshick, R. “Lvis: A dataset for large vocabulary instance segmentation”. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2019. p. 5356–5364, https://www.scopus.com/authid/detail.uri?authorId=35179333300 .
DOI: https://doi.org/10.1109/cvpr.2019.00550.

23. Wu, C., Lin, Z., Cohen, S., Bui, T. & Maji, S. “Phrasecut: Language-based image segmentation in the wild”. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2020. p. 10216–10225. DOI: https://doi.org/10.1109/cvpr42600.2020.01023.

24. Alex, K. “Learning multiple layers of features from tiny images”. – Available from https://www.cs.toronto.edu/kriz/learning-features-2009-TR.pdf. – [Accessed: Jan, 2024].

25. Segment Anything in High Quality – Available from https://github.com/SysCV/sam-hq. – [Accessed: Jan, 2024].

26. SAM 2: Segment Anything in Images and Videos. – Available from https://github.com/facebookresearch/segment-anything. – [Accessed: Jan, 2024].

27. Grounded-Segment-Anything – Available from https://github.com/IDEA-Research/Grounded-Segment-Anything. – [Accessed: Jan, 2024].

28. Fast Segment Anything – Available from https://github.com/CASIA-IVA-Lab/FastSAM. – [Accessed: Jan, 2024].

29. Reis, D., Kupec, J., Hong, J. & Daoudi, A. “Real-time flying object detection with YOLOv8”. *arXiv preprint*. 2023. DOI: https://doi.org/10.48550/arXiv.2305.09972.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Received 13.01.2025
Received after revision 14.03.2025
Accepted 20.03.2025

DOI: https://doi.org/10.15276/hait.08.2025.4
УДК 004:83

Поліпшена модель сегментації для ідентифікації екземплярів об'єктів на основі текстових запитів

Ковтуненко Андрій Романович[1)]
ORCID: http://orcid.org/0009-0004-9072-7779; andrii.kovtunenko@nure.ua. Scopus Author ID: 58362751200
Машталір Сергій Володимирович[1)]
ORCID: http://orcid.org/0000-0002-0917-6622; sergii.mashtalir@nure.ua. Scopus Author ID: 36183980100
[1)] Харківський Національний Університет Радіоелектроніки, пр. Науки 14. Харків, 61166, Україна

АНОТАЦІЯ

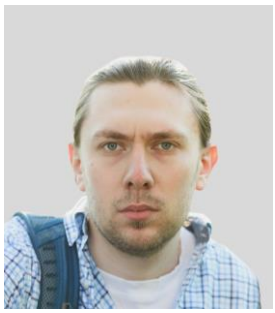
Кількість мультимедійної інформації, що стрімко зросла, вимагає суттєвого розвитку методів її швидкої обробки. При цьому одним із напрямів обробки є попередній аналіз із виділенням характерних ознак зображень для скорочення інформації необхідної для подальших завдань. Одним із видів такого скорочення інформації є сегментація зображень. При цьому загальне завдання сегментації зображень часто зводиться до задачі сегментації об'єктів, що є фундаментальною задачею комп'ютерного зору, що вимагає точного піксельного розмежування об'єктів і розуміння сцени. З розвитком методів обробки природньої мови (NLP) багато підходів були успішно адаптовані до завдань комп'ютерного зору, дозволяючи більш інтуїтивно описувати сцени за допомогою природної мови. На відміну від традиційних моделей, обмежених фіксованим набором класів, підходи на основі обробки природньої мови NLP дозволяють шукати об'єкти на основі атрибутів, що розширює їх застосування. Хоча існуючі методи сегментації об'єктів зазвичай поділяються на одноетапні та двоетапні - залежно від швидкості та точності - залишається прогалина в розробці моделей, які можуть ефективно ідентифікувати та сегментувати об'єкти на основі текстових підказок. Для вирішення цієї проблеми ми пропонуємо модель сегментації екземплярів з необмеженою кількістю класів, здатну виявляти та сегментувати об'єкти за підказками. Наш підхід базується на CLIPSeg, інтегруючи архітектурні модифікації Panoptic-DeepLab та PRN (Panoptic Refinement Network) для прогнозування центрів об'єктів та попіксельних відстаней до меж. На етапі постобробки результати сегментації уточнюються для покращення розділення об'єктів. Запропонована архітектура навчалася на наборах даних LVIS і PhraseCut та оцінюється за допомогою середнього Dice score з сучасними моделями сегментації з відкритими наборами класів. Експериментальні результати показують, що хоча наша модель досягає найвищої швидкості виведення серед методів з відкритими множинами, зберігаючи при цьому якість сегментації на рівні FastSAM, постобробка залишається слабкою ланкою. Майбутні вдосконалення повинні бути спрямовані на усунення самого процесу постобробки або вдосконалення його алгоритму що може призвести до більш ефективної сегментації.

Ключові слова: глибоке навчання; сегментація зображень; згорткові нейронні мережі; архітектури-трансформери; контрастна мовно-образна підготовка; сегментація з нефіксованим набором класів

ABOUT THE AUTHORS

Sergii Volodymyrovych Mashtalir - Doctor of Engineering Science, Professor, Informatics Department. Kharkiv National University of Radio Electronics.14, Nauky Ave. Kharkiv, 61166, Ukraine
ORCID: https://orcid.org/0000-0002-0917-6622; sergii.mashtalir@nure.ua. Scopus Author ID: 36183980100
***Research field*:** Image and video processing; data analysis

Машталір Сергій Володимирович - доктор технічних наук, професор кафедри Інформатики Харківського національного університету радіоелектроніки, проспект Науки 14. Харків, 61166, Україна



Andrii Romanovych Kovtunenko - PhD student, Informatics Department. Kharkiv National University of Radio Electronics.14, Nauky Ave. Kharkiv, 61166, Ukraine
ORCID: https://0009-0004-9072-7779; andrii.kovtunenko@nure.ua. Scopus Author ID: 58362751200
***Research field*:** Image and video processing; data analysis

Ковтуненко Андрій Романович - аспірант кафедри Інформатики Харківського національного університету радіоелектроніки, проспект Науки 14.Харків, 61166, Україна